PUC/Pr Pontifícia Universidade Católica do Paraná
Curso de Bacharelado em Informatica 30/09/2024 - 10:56:57.9
Cadeira de Estruturas de Dados e Arquivos Prof Dr P Kantek
Árvore de cobertura mínima (alg. Kruskal) VIVX634a V: 3.16
75789 ARIEL GUSTAVO RODRIGUES DA S
24PRG502 Exercício : 1 ja vale para P2 de T3 _____ / _____ / _____

Algoritmo de Kruskal: árvore de cobertura mínima

Este problema foi estudado pela primeira vez em 1926 pelo engenheiro Boruvka, quando recebeu a incumbência de eletrificar a Morávia (sudeste da antiga Tchecoslováquia).

Para entender o problema, suponha que você tem uma usina hidrelétrica e um monte de casas espalhadas numa zona rural grande. O objetivo é levar energia elétrica a todas as casas. Ligando todas as casas à usina (ligação em estrela), haverá um grande desperdício de fios e postes.

Outra possibilidade é uma ligação em anel, pela qual são ligadas as casas que estão no perímetro externo da zona, mas permanece a indefinição de como ligar as casas que estão dentro desse perímetro.

Para uma solução ótima (e matematicamente quando se usa o adjetivo ótimo, está se dizendo que é a melhor solução possível), é preciso matematizar o problema, descrevendo-o em termos exatos e passíveis de tratamento matemático. Neste problema, tais termos são o grafo e a matriz de custos. O resultado do problema da ligação é conhecido como árvore de cobertura mínima que nada mais é do que um sub-grafo que conecta todos os vértices e apresenta o menor custo possível.

Dado um grafo não dirigido, cujas arestas comportam custos (distâncias, tempos, volumes, atrasos, postos de pedágio...), este algoritmo gera uma árvore (não há ciclos) que conecta todos os vértices pelo custo mínimo.

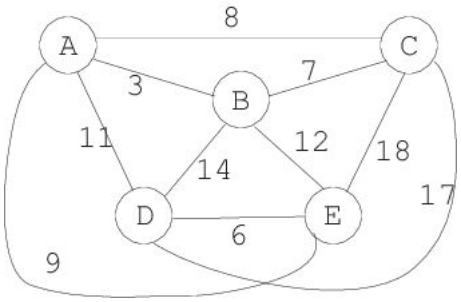
É uma aplicação de "algoritmo guloso"

Suponhamos que o grafo é apresentado através da (diagonal superior da) sua matriz de custos. Por exemplo, seja um grafo de 5 vértices:

Cuja matriz de custos é

	A	B	C	D	E
A	-	3	8	11	9
B	-	-	7	14	12
C	-	-	-	17	18
D	-	-	-	-	6
E	-	-	-	-	-

e cuja figura seja



A etapa 1 do algoritmo é a criação de uma lista de arestas e seus custos

A,B,3 A,C,8 A,D,11 A,E,9 B,C,7
B,D,14 B,E,12 C,D,17 C,E,18 D,E,6

A etapa 2 é ordenar esta lista em ordem crescente de custo e fica:

A,B,3 D,E,6 B,C,7 A,C,8 A,E,9
A,D,11 B,E,12 B,D,14 C,D,17 C,E,18

Etapa 3: Construir um conjunto para cada vértice do grafo

```
1: Inicializar a árvore geradora mínima como vazia
2: enquanto houver mais de um conjunto
3:     Selecionar a aresta de menor custo
4:     se as extremidades estão em conjuntos diferentes
5:         junte os conjuntos e acrescente a aresta na árvore
6:     senão
7:         despreze esta aresta
8:     fim{se}
9: fim{enquanto}
```

Etapa 4: o conjunto (único) contém todos os vértices e a árvore está OK!

Acompanhe o exemplo

- Conjuntos: {A}, {B}, {C}, {D}, {E}, Arvore=[] (vazia)
- Menor: A,B,3. Como A e B estão em conjuntos diferentes, junte-os e fica: {A, B}, {C}, {D}, {E}. Arvore=[(AB)].
- Menor: D,E,6. Como D e E... Fica: {A,B}, {C}, {D,E}. Arvore=[(AB), (DE)]
- menor: B,C,7. Como B e C... Fica {A,B,C}, {D,E}. Arvore=[(AB), (DE), (BC)]
- menor: A,C,8. Como A e C já são do mesmo conjunto, a aresta é desprezada
- menor: A,E,9. Como A e E estão em dois conjuntos, fica:
- {A,B,C,D,E}. Arvore=[(AB), (DE), (BC), (AE)] com um custo de 25 e acabou o algoritmo.

Outro exemplo

Acompanhe este exemplo: Seja o grafo 8 x 8:

	1	2	3	4	5	6	7	8
1	0	7	22	38	64	38	2	37
2	0	0	18	46	44	11	17	52
3	0	0	0	56	43	49	33	59
4	0	0	0	0	44	43	7	43
5	0	0	0	0	0	26	36	45
6	0	0	0	0	0	0	60	47
7	0	0	0	0	0	0	0	44
8	0	0	0	0	0	0	0	0

Que gera a seguinte lista de arestas (**já ordenada**):

CS	OR	DE	CS	OR	DE	CS	OR	DE	CS	OR	DE
==	==	==	==	==	==	==	==	==	==	==	==
2	1	7	26	5	6	43	4	6**	47	6	8
7	1	2**	33	3	7	43	4	8**	49	3	6
7	4	7**	36	5	7	44	2	5**	52	2	8
11	2	6	37	1	8	44	4	5**	56	3	4
17	2	7	38	1	4**	44	7	8**	59	3	8
18	2	3	38	1	6**	45	5	8	60	6	7
22	1	3	43	3	5**	46	2	4	64	1	5

Observação importante: Note as arestas marcadas com **. Elas estão empatadas. Observe como, neste caso o algoritmo preserva a ordem de construção da tabela.

Esta lista quando aplicada no algoritmo gera a seguinte árvore mínima:

```
1,7   1,2   4,7   2,6   2,3   5,6   1,8;
com um custo mínimo de 108
```

☞ Para você fazer

1.

	1	2	3	4	5	6	7	8	9	10
1	0	50	99	58	8	71	19	31	21	72
2	0	0	21	29	13	87	81	15	31	55
3	0	0	0	69	10	30	99	14	34	66
4	0	0	0	0	93	94	19	38	58	89
5	0	0	0	0	0	73	93	95	76	96
6	0	0	0	0	0	0	67	33	11	49
7	0	0	0	0	0	0	0	49	47	96
8	0	0	0	0	0	0	0	0	31	6
9	0	0	0	0	0	0	0	0	0	86
10	0	0	0	0	0	0	0	0	0	0

Pede-se a ÚLTIMA aresta que forma a árvore mínima: de_____a _____,

Pede-se também o CUSTO TOTAL DA ÁRVORE DE COBERTURA MÍNIMA: _____

2.

	1	2	3	4	5	6	7	8	9	10
1	0	15	5	93	17	71	34	12	17	89
2	0	0	4	75	66	42	95	87	57	14
3	0	0	0	40	90	95	73	16	14	49
4	0	0	0	0	91	44	7	59	88	66
5	0	0	0	0	0	68	73	18	9	96
6	0	0	0	0	0	0	7	39	79	49
7	0	0	0	0	0	0	0	40	95	78
8	0	0	0	0	0	0	0	0	93	30
9	0	0	0	0	0	0	0	0	0	82
10	0	0	0	0	0	0	0	0	0	0

Pede-se a ÚLTIMA aresta que forma a árvore mínima: de_____a _____,

Pede-se também o CUSTO TOTAL DA ÁRVORE DE COBERTURA MÍNIMA: _____

PUC/Pr Pontifícia Universidade Católica do Paraná
Curso de Bacharelado em Informatica 30/09/2024 - 10:56:57.9
Cadeira de Estruturas de Dados e Arquivos Prof Dr P Kantek
Árvore de cobertura mínima (alg. Kruskal) VIVX634a V: 3.16
75796 CASSIANO MAGNO CHAGAS E SA
24PRG502 Exercício : 2 ja vale para P2 de T3 _____ / _____ / _____

Algoritmo de Kruskal: árvore de cobertura mínima

Este problema foi estudado pela primeira vez em 1926 pelo engenheiro Boruvka, quando recebeu a incumbência de eletrificar a Morávia (sudeste da antiga Tchecoslováquia).

Para entender o problema, suponha que você tem uma usina hidrelétrica e um monte de casas espalhadas numa zona rural grande. O objetivo é levar energia elétrica a todas as casas. Ligando todas as casas à usina (ligação em estrela), haverá um grande desperdício de fios e postes.

Outra possibilidade é uma ligação em anel, pela qual são ligadas as casas que estão no perímetro externo da zona, mas permanece a indefinição de como ligar as casas que estão dentro desse perímetro.

Para uma solução ótima (e matematicamente quando se usa o adjetivo ótimo, está se dizendo que é a melhor solução possível), é preciso matematizar o problema, descrevendo-o em termos exatos e passíveis de tratamento matemático. Neste problema, tais termos são o grafo e a matriz de custos. O resultado do problema da ligação é conhecido como árvore de cobertura mínima que nada mais é do que um sub-grafo que conecta todos os vértices e apresenta o menor custo possível.

Dado um grafo não dirigido, cujas arestas comportam custos (distâncias, tempos, volumes, atrasos, postos de pedágio...), este algoritmo gera uma árvore (não há ciclos) que conecta todos os vértices pelo custo mínimo.

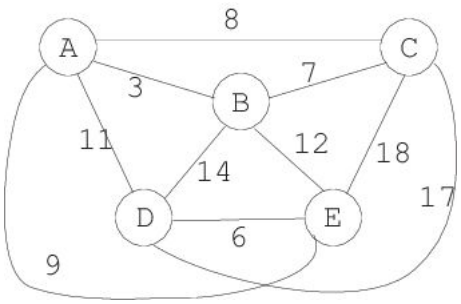
É uma aplicação de "algoritmo guloso"

Suponhamos que o grafo é apresentado através da (diagonal superior da) sua matriz de custos. Por exemplo, seja um grafo de 5 vértices:

Cuja matriz de custos é

	A	B	C	D	E
A	-	3	8	11	9
B	-	-	7	14	12
C	-	-	-	17	18
D	-	-	-	-	6
E	-	-	-	-	-

e cuja figura seja



A etapa 1 do algoritmo é a criação de uma lista de arestas e seus custos

A,B,3 A,C,8 A,D,11 A,E,9 B,C,7
B,D,14 B,E,12 C,D,17 C,E,18 D,E,6

A etapa 2 é ordenar esta lista em ordem crescente de custo e fica:

A,B,3 D,E,6 B,C,7 A,C,8 A,E,9
A,D,11 B,E,12 B,D,14 C,D,17 C,E,18

Etapa 3: Construir um conjunto para cada vértice do grafo

```
1: Inicializar a árvore geradora mínima como vazia
2: enquanto houver mais de um conjunto
3:     Selecionar a aresta de menor custo
4:     se as extremidades estão em conjuntos diferentes
5:         junte os conjuntos e acrescente a aresta na árvore
6:     senão
7:         despreze esta aresta
8:     fim{se}
9: fim{enquanto}
```

Etapa 4: o conjunto (único) contém todos os vértices e a árvore está OK!

Acompanhe o exemplo

- Conjuntos: {A}, {B}, {C}, {D}, {E}, Arvore=[] (vazia)
- Menor: A,B,3. Como A e B estão em conjuntos diferentes, junte-os e fica: {A, B}, {C}, {D}, {E}. Arvore=[(AB)].
- Menor: D,E,6. Como D e E... Fica: {A,B}, {C}, {D,E}. Arvore=[(AB), (DE)]
- menor: B,C,7. Como B e C... Fica {A,B,C}, {D,E}. Arvore=[(AB), (DE), (BC)]
- menor: A,C,8. Como A e C já são do mesmo conjunto, a aresta é desprezada
- menor: A,E,9. Como A e E estão em dois conjuntos, fica:
- {A,B,C,D,E}. Arvore=[(AB), (DE), (BC), (AE)] com um custo de 25 e acabou o algoritmo.

Outro exemplo

Acompanhe este exemplo: Seja o grafo 8 x 8:

	1	2	3	4	5	6	7	8
1	0	7	22	38	64	38	2	37
2	0	0	18	46	44	11	17	52
3	0	0	0	56	43	49	33	59
4	0	0	0	0	44	43	7	43
5	0	0	0	0	0	26	36	45
6	0	0	0	0	0	0	60	47
7	0	0	0	0	0	0	0	44
8	0	0	0	0	0	0	0	0

Que gera a seguinte lista de arestas (**já ordenada**):

CS	OR	DE	CS	OR	DE	CS	OR	DE	CS	OR	DE
==	==	==	==	==	==	==	==	==	==	==	==
2	1	7	26	5	6	43	4	6**	47	6	8
7	1	2**	33	3	7	43	4	8**	49	3	6
7	4	7**	36	5	7	44	2	5**	52	2	8
11	2	6	37	1	8	44	4	5**	56	3	4
17	2	7	38	1	4**	44	7	8**	59	3	8
18	2	3	38	1	6**	45	5	8	60	6	7
22	1	3	43	3	5**	46	2	4	64	1	5

Observação importante: Note as arestas marcadas com **. Elas estão empatadas. Observe como, neste caso o algoritmo preserva a ordem de construção da tabela.

Esta lista quando aplicada no algoritmo gera a seguinte árvore mínima:

```
1,7   1,2   4,7   2,6   2,3   5,6   1,8;
com um custo mínimo de 108
```

☞ Para você fazer

1.

	1	2	3	4	5	6	7	8	9	10
1	0	52	97	74	80	36	94	85	17	48
2	0	0	41	50	72	28	87	65	58	4
3	0	0	0	3	28	49	99	78	38	88
4	0	0	0	0	3	75	80	96	21	5
5	0	0	0	0	0	6	78	91	42	77
6	0	0	0	0	0	0	88	8	59	65
7	0	0	0	0	0	0	0	73	87	27
8	0	0	0	0	0	0	0	0	57	5
9	0	0	0	0	0	0	0	0	0	67
10	0	0	0	0	0	0	0	0	0	0

Pede-se a ÚLTIMA aresta que forma a árvore mínima: de_____a _____,

Pede-se também o CUSTO TOTAL DA ÁRVORE DE COBERTURA MÍNIMA: _____

2.

	1	2	3	4	5	6	7	8	9	10
1	0	66	92	9	35	67	79	37	81	76
2	0	0	48	41	92	30	1	88	35	97
3	0	0	0	60	61	29	66	31	37	73
4	0	0	0	0	71	50	99	23	51	94
5	0	0	0	0	0	68	43	47	6	34
6	0	0	0	0	0	0	97	27	50	83
7	0	0	0	0	0	0	0	24	57	3
8	0	0	0	0	0	0	0	0	77	42
9	0	0	0	0	0	0	0	0	0	94
10	0	0	0	0	0	0	0	0	0	0

Pede-se a ÚLTIMA aresta que forma a árvore mínima: de_____a _____,

Pede-se também o CUSTO TOTAL DA ÁRVORE DE COBERTURA MÍNIMA: _____

PUC/Pr Pontifícia Universidade Católica do Paraná
Curso de Bacharelado em Informatica 30/09/2024 - 10:56:57.9
Cadeira de Estruturas de Dados e Arquivos Prof Dr P Kantek
Árvore de cobertura mínima (alg. Kruskal) VIVX634a V: 3.16
75808 CHRISTOPHER BUCH FILIPAK
24PRG502 Exercício : 3 ja vale para P2 de T3 _____ / _____ / _____

Algoritmo de Kruskal: árvore de cobertura mínima

Este problema foi estudado pela primeira vez em 1926 pelo engenheiro Boruvka, quando recebeu a incumbência de eletrificar a Morávia (sudeste da antiga Tchecoslováquia).

Para entender o problema, suponha que você tem uma usina hidrelétrica e um monte de casas espalhadas numa zona rural grande. O objetivo é levar energia elétrica a todas as casas. Ligando todas as casas à usina (ligação em estrela), haverá um grande desperdício de fios e postes.

Outra possibilidade é uma ligação em anel, pela qual são ligadas as casas que estão no perímetro externo da zona, mas permanece a indefinição de como ligar as casas que estão dentro desse perímetro.

Para uma solução ótima (e matematicamente quando se usa o adjetivo ótimo, está se dizendo que é a melhor solução possível), é preciso matematizar o problema, descrevendo-o em termos exatos e passíveis de tratamento matemático. Neste problema, tais termos são o grafo e a matriz de custos. O resultado do problema da ligação é conhecido como árvore de cobertura mínima que nada mais é do que um sub-grafo que conecta todos os vértices e apresenta o menor custo possível.

Dado um grafo não dirigido, cujas arestas comportam custos (distâncias, tempos, volumes, atrasos, postos de pedágio...), este algoritmo gera uma árvore (não há ciclos) que conecta todos os vértices pelo custo mínimo.

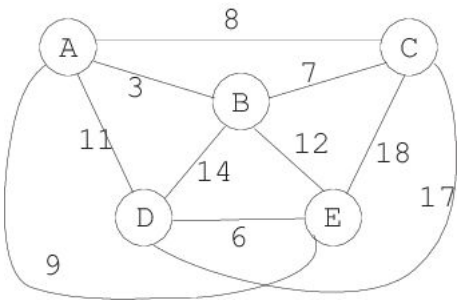
É uma aplicação de "algoritmo guloso"

Suponhamos que o grafo é apresentado através da (diagonal superior da) sua matriz de custos. Por exemplo, seja um grafo de 5 vértices:

Cuja matriz de custos é

	A	B	C	D	E
A	-	3	8	11	9
B	-	-	7	14	12
C	-	-	-	17	18
D	-	-	-	-	6
E	-	-	-	-	-

e cuja figura seja



A etapa 1 do algoritmo é a criação de uma lista de arestas e seus custos

A,B,3 A,C,8 A,D,11 A,E,9 B,C,7
B,D,14 B,E,12 C,D,17 C,E,18 D,E,6

A etapa 2 é ordenar esta lista em ordem crescente de custo e fica:

A,B,3 D,E,6 B,C,7 A,C,8 A,E,9
A,D,11 B,E,12 B,D,14 C,D,17 C,E,18

Etapa 3: Construir um conjunto para cada vértice do grafo

```
1: Inicializar a árvore geradora mínima como vazia
2: enquanto houver mais de um conjunto
3:     Selecionar a aresta de menor custo
4:     se as extremidades estão em conjuntos diferentes
5:         junte os conjuntos e acrescente a aresta na árvore
6:     senão
7:         despreze esta aresta
8:     fim{se}
9: fim{enquanto}
```

Etapa 4: o conjunto (único) contém todos os vértices e a árvore está OK!

Acompanhe o exemplo

- Conjuntos: {A}, {B}, {C}, {D}, {E}, Arvore=[] (vazia)
- Menor: A,B,3. Como A e B estão em conjuntos diferentes, junte-os e fica: {A, B}, {C}, {D}, {E}. Arvore=[(AB)].
- Menor: D,E,6. Como D e E... Fica: {A,B}, {C}, {D,E}. Arvore=[(AB), (DE)]
- menor: B,C,7. Como B e C... Fica {A,B,C}, {D,E}. Arvore=[(AB), (DE), (BC)]
- menor: A,C,8. Como A e C já são do mesmo conjunto, a aresta é desprezada
- menor: A,E,9. Como A e E estão em dois conjuntos, fica:
- {A,B,C,D,E}. Arvore=[(AB), (DE), (BC), (AE)] com um custo de 25 e acabou o algoritmo.

Outro exemplo

Acompanhe este exemplo: Seja o grafo 8 x 8:

	1	2	3	4	5	6	7	8
1	0	7	22	38	64	38	2	37
2	0	0	18	46	44	11	17	52
3	0	0	0	56	43	49	33	59
4	0	0	0	0	44	43	7	43
5	0	0	0	0	0	26	36	45
6	0	0	0	0	0	0	60	47
7	0	0	0	0	0	0	0	44
8	0	0	0	0	0	0	0	0

Que gera a seguinte lista de arestas (**já ordenada**):

CS	OR	DE	CS	OR	DE	CS	OR	DE	CS	OR	DE
==	==	==	==	==	==	==	==	==	==	==	==
2	1	7	26	5	6	43	4	6**	47	6	8
7	1	2**	33	3	7	43	4	8**	49	3	6
7	4	7**	36	5	7	44	2	5**	52	2	8
11	2	6	37	1	8	44	4	5**	56	3	4
17	2	7	38	1	4**	44	7	8**	59	3	8
18	2	3	38	1	6**	45	5	8	60	6	7
22	1	3	43	3	5**	46	2	4	64	1	5

Observação importante: Note as arestas marcadas com **. Elas estão empatadas. Observe como, neste caso o algoritmo preserva a ordem de construção da tabela.

Esta lista quando aplicada no algoritmo gera a seguinte árvore mínima:

```
1,7   1,2   4,7   2,6   2,3   5,6   1,8;
com um custo mínimo de 108
```

☞ Para você fazer

1.

	1	2	3	4	5	6	7	8	9	10
1	0	42	65	86	54	78	18	55	74	10
2	0	0	11	27	31	57	21	18	29	29
3	0	0	0	62	99	27	68	3	17	52
4	0	0	0	0	87	88	17	75	54	55
5	0	0	0	0	0	6	45	60	48	72
6	0	0	0	0	0	0	73	83	67	31
7	0	0	0	0	0	0	0	69	96	17
8	0	0	0	0	0	0	0	0	43	45
9	0	0	0	0	0	0	0	0	0	95
10	0	0	0	0	0	0	0	0	0	0

Pede-se a ÚLTIMA aresta que forma a árvore mínima: de_____a _____,

Pede-se também o CUSTO TOTAL DA ÁRVORE DE COBERTURA MÍNIMA: _____

2.

	1	2	3	4	5	6	7	8	9	10
1	0	32	6	50	97	97	44	74	31	35
2	0	0	36	37	74	31	55	12	31	86
3	0	0	0	49	52	1	1	47	52	34
4	0	0	0	0	50	42	12	24	96	13
5	0	0	0	0	0	96	92	74	81	64
6	0	0	0	0	0	0	35	2	13	87
7	0	0	0	0	0	0	0	15	47	15
8	0	0	0	0	0	0	0	0	74	29
9	0	0	0	0	0	0	0	0	0	67
10	0	0	0	0	0	0	0	0	0	0

Pede-se a ÚLTIMA aresta que forma a árvore mínima: de_____a _____,

Pede-se também o CUSTO TOTAL DA ÁRVORE DE COBERTURA MÍNIMA: _____

PUC/Pr Pontifícia Universidade Católica do Paraná
Curso de Bacharelado em Informatica 30/09/2024 - 10:56:57.9
Cadeira de Estruturas de Dados e Arquivos Prof Dr P Kantek
Árvore de cobertura mínima (alg. Kruskal) VIVX634a V: 3.16
75815 ERAN MARTINEZ RAMOS
24PRG502 Exercício : 4 ja vale para P2 de T3 _____ / _____ / _____

Algoritmo de Kruskal: árvore de cobertura mínima

Este problema foi estudado pela primeira vez em 1926 pelo engenheiro Boruvka, quando recebeu a incumbência de eletrificar a Morávia (sudeste da antiga Tchecoslováquia).

Para entender o problema, suponha que você tem uma usina hidrelétrica e um monte de casas espalhadas numa zona rural grande. O objetivo é levar energia elétrica a todas as casas. Ligando todas as casas à usina (ligação em estrela), haverá um grande desperdício de fios e postes.

Outra possibilidade é uma ligação em anel, pela qual são ligadas as casas que estão no perímetro externo da zona, mas permanece a indefinição de como ligar as casas que estão dentro desse perímetro.

Para uma solução ótima (e matematicamente quando se usa o adjetivo ótimo, está se dizendo que é a melhor solução possível), é preciso matematizar o problema, descrevendo-o em termos exatos e passíveis de tratamento matemático. Neste problema, tais termos são o grafo e a matriz de custos. O resultado do problema da ligação é conhecido como árvore de cobertura mínima que nada mais é do que um sub-grafo que conecta todos os vértices e apresenta o menor custo possível.

Dado um grafo não dirigido, cujas arestas comportam custos (distâncias, tempos, volumes, atrasos, postos de pedágio...), este algoritmo gera uma árvore (não há ciclos) que conecta todos os vértices pelo custo mínimo.

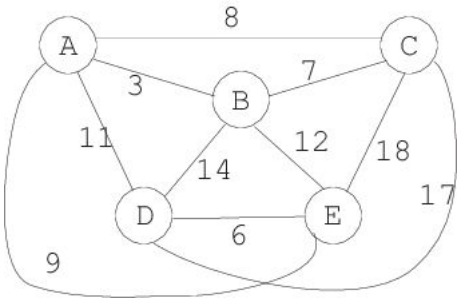
É uma aplicação de "algoritmo guloso"

Suponhamos que o grafo é apresentado através da (diagonal superior da) sua matriz de custos. Por exemplo, seja um grafo de 5 vértices:

Cuja matriz de custos é

	A	B	C	D	E
A	-	3	8	11	9
B	-	-	7	14	12
C	-	-	-	17	18
D	-	-	-	-	6
E	-	-	-	-	-

e cuja figura seja



A etapa 1 do algoritmo é a criação de uma lista de arestas e seus custos

A,B,3 A,C,8 A,D,11 A,E,9 B,C,7
B,D,14 B,E,12 C,D,17 C,E,18 D,E,6

A etapa 2 é ordenar esta lista em ordem crescente de custo e fica:

A,B,3 D,E,6 B,C,7 A,C,8 A,E,9
A,D,11 B,E,12 B,D,14 C,D,17 C,E,18

Etapa 3: Construir um conjunto para cada vértice do grafo

```
1: Inicializar a árvore geradora mínima como vazia
2: enquanto houver mais de um conjunto
3:     Selecionar a aresta de menor custo
4:     se as extremidades estão em conjuntos diferentes
5:         junte os conjuntos e acrescente a aresta na árvore
6:     senão
7:         despreze esta aresta
8:     fim{se}
9: fim{enquanto}
```

Etapa 4: o conjunto (único) contém todos os vértices e a árvore está OK!

Acompanhe o exemplo

- Conjuntos: {A}, {B}, {C}, {D}, {E}, Arvore=[] (vazia)
- Menor: A,B,3. Como A e B estão em conjuntos diferentes, junte-os e fica: {A, B}, {C}, {D}, {E}. Arvore=[(AB)].
- Menor: D,E,6. Como D e E... Fica: {A,B}, {C}, {D,E}. Arvore=[(AB), (DE)]
- menor: B,C,7. Como B e C... Fica {A,B,C}, {D,E}. Arvore=[(AB), (DE), (BC)]
- menor: A,C,8. Como A e C já são do mesmo conjunto, a aresta é desprezada
- menor: A,E,9. Como A e E estão em dois conjuntos, fica:
- {A,B,C,D,E}. Arvore=[(AB), (DE), (BC), (AE)] com um custo de 25 e acabou o algoritmo.

Outro exemplo

Acompanhe este exemplo: Seja o grafo 8 x 8:

	1	2	3	4	5	6	7	8
1	0	7	22	38	64	38	2	37
2	0	0	18	46	44	11	17	52
3	0	0	0	56	43	49	33	59
4	0	0	0	0	44	43	7	43
5	0	0	0	0	0	26	36	45
6	0	0	0	0	0	0	60	47
7	0	0	0	0	0	0	0	44
8	0	0	0	0	0	0	0	0

Que gera a seguinte lista de arestas (**já ordenada**):

CS	OR	DE	CS	OR	DE	CS	OR	DE	CS	OR	DE
==	==	==	==	==	==	==	==	==	==	==	==
2	1	7	26	5	6	43	4	6**	47	6	8
7	1	2**	33	3	7	43	4	8**	49	3	6
7	4	7**	36	5	7	44	2	5**	52	2	8
11	2	6	37	1	8	44	4	5**	56	3	4
17	2	7	38	1	4**	44	7	8**	59	3	8
18	2	3	38	1	6**	45	5	8	60	6	7
22	1	3	43	3	5**	46	2	4	64	1	5

Observação importante: Note as arestas marcadas com **. Elas estão empatadas. Observe como, neste caso o algoritmo preserva a ordem de construção da tabela.

Esta lista quando aplicada no algoritmo gera a seguinte árvore mínima:

```
1,7   1,2   4,7   2,6   2,3   5,6   1,8;
com um custo mínimo de 108
```

☞ Para você fazer

1.

	1	2	3	4	5	6	7	8	9	10
1	0	94	8	15	72	79	6	91	60	92
2	0	0	90	13	14	68	83	27	60	89
3	0	0	0	68	45	44	48	59	11	47
4	0	0	0	0	39	90	60	3	49	39
5	0	0	0	0	0	72	35	14	34	11
6	0	0	0	0	0	0	26	72	13	64
7	0	0	0	0	0	0	0	59	45	81
8	0	0	0	0	0	0	0	0	38	23
9	0	0	0	0	0	0	0	0	0	51
10	0	0	0	0	0	0	0	0	0	0

Pede-se a ÚLTIMA aresta que forma a árvore mínima: de_____a _____,

Pede-se também o CUSTO TOTAL DA ÁRVORE DE COBERTURA MÍNIMA: _____

2.

	1	2	3	4	5	6	7	8	9	10
1	0	10	18	10	12	92	98	15	43	27
2	0	0	97	61	19	74	79	17	21	21
3	0	0	0	92	85	47	26	34	7	44
4	0	0	0	0	16	41	45	85	9	73
5	0	0	0	0	0	65	62	91	71	2
6	0	0	0	0	0	0	68	44	31	36
7	0	0	0	0	0	0	0	94	77	22
8	0	0	0	0	0	0	0	0	68	39
9	0	0	0	0	0	0	0	0	0	6
10	0	0	0	0	0	0	0	0	0	0

Pede-se a ÚLTIMA aresta que forma a árvore mínima: de_____a _____,

Pede-se também o CUSTO TOTAL DA ÁRVORE DE COBERTURA MÍNIMA: _____

PUC/Pr Pontifícia Universidade Católica do Paraná
Curso de Bacharelado em Informatica 30/09/2024 - 10:56:57.9
Cadeira de Estruturas de Dados e Arquivos Prof Dr P Kantek
Árvore de cobertura mínima (alg. Kruskal) VIVX634a V: 3.16
75822 HENRIQUE IVANKIO GUSSE
24PRG502 Exercício : 5 ja vale para P2 de T3 _____ / _____ / _____

Algoritmo de Kruskal: árvore de cobertura mínima

Este problema foi estudado pela primeira vez em 1926 pelo engenheiro Boruvka, quando recebeu a incumbência de eletrificar a Morávia (sudeste da antiga Tchecoslováquia).

Para entender o problema, suponha que você tem uma usina hidrelétrica e um monte de casas espalhadas numa zona rural grande. O objetivo é levar energia elétrica a todas as casas. Ligando todas as casas à usina (ligação em estrela), haverá um grande desperdício de fios e postes.

Outra possibilidade é uma ligação em anel, pela qual são ligadas as casas que estão no perímetro externo da zona, mas permanece a indefinição de como ligar as casas que estão dentro desse perímetro.

Para uma solução ótima (e matematicamente quando se usa o adjetivo ótimo, está se dizendo que é a melhor solução possível), é preciso matematizar o problema, descrevendo-o em termos exatos e passíveis de tratamento matemático. Neste problema, tais termos são o grafo e a matriz de custos. O resultado do problema da ligação é conhecido como árvore de cobertura mínima que nada mais é do que um sub-grafo que conecta todos os vértices e apresenta o menor custo possível.

Dado um grafo não dirigido, cujas arestas comportam custos (distâncias, tempos, volumes, atrasos, postos de pedágio...), este algoritmo gera uma árvore (não há ciclos) que conecta todos os vértices pelo custo mínimo.

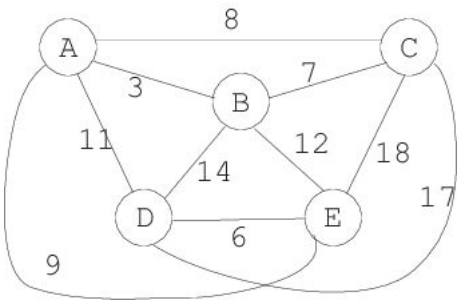
É uma aplicação de "algoritmo guloso"

Suponhamos que o grafo é apresentado através da (diagonal superior da) sua matriz de custos. Por exemplo, seja um grafo de 5 vértices:

Cuja matriz de custos é

	A	B	C	D	E
A	-	3	8	11	9
B	-	-	7	14	12
C	-	-	-	17	18
D	-	-	-	-	6
E	-	-	-	-	-

e cuja figura seja



A etapa 1 do algoritmo é a criação de uma lista de arestas e seus custos

A,B,3 A,C,8 A,D,11 A,E,9 B,C,7
B,D,14 B,E,12 C,D,17 C,E,18 D,E,6

A etapa 2 é ordenar esta lista em ordem crescente de custo e fica:

A,B,3 D,E,6 B,C,7 A,C,8 A,E,9
A,D,11 B,E,12 B,D,14 C,D,17 C,E,18

Etapa 3: Construir um conjunto para cada vértice do grafo

```
1: Inicializar a árvore geradora mínima como vazia
2: enquanto houver mais de um conjunto
3:     Selecionar a aresta de menor custo
4:     se as extremidades estão em conjuntos diferentes
5:         junte os conjuntos e acrescente a aresta na árvore
6:     senão
7:         despreze esta aresta
8:     fim{se}
9: fim{enquanto}
```

Etapa 4: o conjunto (único) contém todos os vértices e a árvore está OK!

Acompanhe o exemplo

- Conjuntos: {A}, {B}, {C}, {D}, {E}, Arvore=[] (vazia)
- Menor: A,B,3. Como A e B estão em conjuntos diferentes, junte-os e fica: {A, B}, {C}, {D}, {E}. Arvore=[(AB)].
- Menor: D,E,6. Como D e E... Fica: {A,B}, {C}, {D,E}. Arvore=[(AB), (DE)]
- menor: B,C,7. Como B e C... Fica {A,B,C}, {D,E}. Arvore=[(AB), (DE), (BC)]
- menor: A,C,8. Como A e C já são do mesmo conjunto, a aresta é desprezada
- menor: A,E,9. Como A e E estão em dois conjuntos, fica:
- {A,B,C,D,E}. Arvore=[(AB), (DE), (BC), (AE)] com um custo de 25 e acabou o algoritmo.

Outro exemplo

Acompanhe este exemplo: Seja o grafo 8 x 8:

	1	2	3	4	5	6	7	8
1	0	7	22	38	64	38	2	37
2	0	0	18	46	44	11	17	52
3	0	0	0	56	43	49	33	59
4	0	0	0	0	44	43	7	43
5	0	0	0	0	0	26	36	45
6	0	0	0	0	0	0	60	47
7	0	0	0	0	0	0	0	44
8	0	0	0	0	0	0	0	0

Que gera a seguinte lista de arestas (**já ordenada**):

CS	OR	DE	CS	OR	DE	CS	OR	DE	CS	OR	DE
==	==	==	==	==	==	==	==	==	==	==	==
2	1	7	26	5	6	43	4	6**	47	6	8
7	1	2**	33	3	7	43	4	8**	49	3	6
7	4	7**	36	5	7	44	2	5**	52	2	8
11	2	6	37	1	8	44	4	5**	56	3	4
17	2	7	38	1	4**	44	7	8**	59	3	8
18	2	3	38	1	6**	45	5	8	60	6	7
22	1	3	43	3	5**	46	2	4	64	1	5

Observação importante: Note as arestas marcadas com **. Elas estão empatadas. Observe como, neste caso o algoritmo preserva a ordem de construção da tabela.

Esta lista quando aplicada no algoritmo gera a seguinte árvore mínima:

```
1,7   1,2   4,7   2,6   2,3   5,6   1,8;
com um custo mínimo de 108
```

☞ Para você fazer

1.

	1	2	3	4	5	6	7	8	9	10
1	0	79	93	95	22	95	30	93	30	54
2	0	0	34	23	90	17	50	92	8	36
3	0	0	0	92	60	50	92	95	48	50
4	0	0	0	0	41	79	75	2	37	75
5	0	0	0	0	0	26	99	24	24	81
6	0	0	0	0	0	0	35	47	57	22
7	0	0	0	0	0	0	0	3	42	48
8	0	0	0	0	0	0	0	0	30	33
9	0	0	0	0	0	0	0	0	0	80
10	0	0	0	0	0	0	0	0	0	0

Pede-se a ÚLTIMA aresta que forma a árvore mínima: de_____a _____,

Pede-se também o CUSTO TOTAL DA ÁRVORE DE COBERTURA MÍNIMA: _____

2.

	1	2	3	4	5	6	7	8	9	10
1	0	19	99	5	86	88	44	53	54	3
2	0	0	26	74	94	8	79	44	39	40
3	0	0	0	43	51	41	46	28	27	97
4	0	0	0	0	18	81	88	21	94	87
5	0	0	0	0	0	90	95	52	86	76
6	0	0	0	0	0	0	49	95	30	43
7	0	0	0	0	0	0	0	47	53	90
8	0	0	0	0	0	0	0	0	33	15
9	0	0	0	0	0	0	0	0	0	3
10	0	0	0	0	0	0	0	0	0	0

Pede-se a ÚLTIMA aresta que forma a árvore mínima: de_____a _____,

Pede-se também o CUSTO TOTAL DA ÁRVORE DE COBERTURA MÍNIMA: _____

PUC/Pr Pontifícia Universidade Católica do Paraná
Curso de Bacharelado em Informatica 30/09/2024 - 10:56:57.9
Cadeira de Estruturas de Dados e Arquivos Prof Dr P Kantek
Árvore de cobertura mínima (alg. Kruskal) VIVX634a V: 3.16
75839 IAN DOLABELLA OHATA
24PRG502 Exercício : 6 ja vale para P2 de T3 _____ / _____ / _____

Algoritmo de Kruskal: árvore de cobertura mínima

Este problema foi estudado pela primeira vez em 1926 pelo engenheiro Boruvka, quando recebeu a incumbência de eletrificar a Morávia (sudeste da antiga Tchecoslováquia).

Para entender o problema, suponha que você tem uma usina hidrelétrica e um monte de casas espalhadas numa zona rural grande. O objetivo é levar energia elétrica a todas as casas. Ligando todas as casas à usina (ligação em estrela), haverá um grande desperdício de fios e postes.

Outra possibilidade é uma ligação em anel, pela qual são ligadas as casas que estão no perímetro externo da zona, mas permanece a indefinição de como ligar as casas que estão dentro desse perímetro.

Para uma solução ótima (e matematicamente quando se usa o adjetivo ótimo, está se dizendo que é a melhor solução possível), é preciso matematizar o problema, descrevendo-o em termos exatos e passíveis de tratamento matemático. Neste problema, tais termos são o grafo e a matriz de custos. O resultado do problema da ligação é conhecido como árvore de cobertura mínima que nada mais é do que um sub-grafo que conecta todos os vértices e apresenta o menor custo possível.

Dado um grafo não dirigido, cujas arestas comportam custos (distâncias, tempos, volumes, atrasos, postos de pedágio...), este algoritmo gera uma árvore (não há ciclos) que conecta todos os vértices pelo custo mínimo.

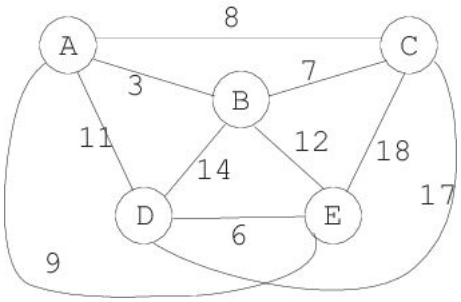
É uma aplicação de "algoritmo guloso"

Suponhamos que o grafo é apresentado através da (diagonal superior da) sua matriz de custos. Por exemplo, seja um grafo de 5 vértices:

Cuja matriz de custos é

	A	B	C	D	E
A	-	3	8	11	9
B	-	-	7	14	12
C	-	-	-	17	18
D	-	-	-	-	6
E	-	-	-	-	-

e cuja figura seja



A etapa 1 do algoritmo é a criação de uma lista de arestas e seus custos

A,B,3 A,C,8 A,D,11 A,E,9 B,C,7
B,D,14 B,E,12 C,D,17 C,E,18 D,E,6

A etapa 2 é ordenar esta lista em ordem crescente de custo e fica:

A,B,3 D,E,6 B,C,7 A,C,8 A,E,9
A,D,11 B,E,12 B,D,14 C,D,17 C,E,18

Etapa 3: Construir um conjunto para cada vértice do grafo

```
1: Inicializar a árvore geradora mínima como vazia
2: enquanto houver mais de um conjunto
3:     Selecionar a aresta de menor custo
4:     se as extremidades estão em conjuntos diferentes
5:         junte os conjuntos e acrescente a aresta na árvore
6:     senão
7:         despreze esta aresta
8:     fim{se}
9: fim{enquanto}
```

Etapa 4: o conjunto (único) contém todos os vértices e a árvore está OK!

Acompanhe o exemplo

- Conjuntos: {A}, {B}, {C}, {D}, {E}, Arvore=[] (vazia)
- Menor: A,B,3. Como A e B estão em conjuntos diferentes, junte-os e fica: {A, B}, {C}, {D}, {E}. Arvore=[(AB)].
- Menor: D,E,6. Como D e E... Fica: {A,B}, {C}, {D,E}. Arvore=[(AB), (DE)]
- menor: B,C,7. Como B e C... Fica {A,B,C}, {D,E}. Arvore=[(AB), (DE), (BC)]
- menor: A,C,8. Como A e C já são do mesmo conjunto, a aresta é desprezada
- menor: A,E,9. Como A e E estão em dois conjuntos, fica:
- {A,B,C,D,E}. Arvore=[(AB), (DE), (BC), (AE)] com um custo de 25 e acabou o algoritmo.

Outro exemplo

Acompanhe este exemplo: Seja o grafo 8 x 8:

	1	2	3	4	5	6	7	8
1	0	7	22	38	64	38	2	37
2	0	0	18	46	44	11	17	52
3	0	0	0	56	43	49	33	59
4	0	0	0	0	44	43	7	43
5	0	0	0	0	0	26	36	45
6	0	0	0	0	0	0	60	47
7	0	0	0	0	0	0	0	44
8	0	0	0	0	0	0	0	0

Que gera a seguinte lista de arestas (**já ordenada**):

CS	OR	DE	CS	OR	DE	CS	OR	DE	CS	OR	DE
==	==	==	==	==	==	==	==	==	==	==	==
2	1	7	26	5	6	43	4	6**	47	6	8
7	1	2**	33	3	7	43	4	8**	49	3	6
7	4	7**	36	5	7	44	2	5**	52	2	8
11	2	6	37	1	8	44	4	5**	56	3	4
17	2	7	38	1	4**	44	7	8**	59	3	8
18	2	3	38	1	6**	45	5	8	60	6	7
22	1	3	43	3	5**	46	2	4	64	1	5

Observação importante: Note as arestas marcadas com **. Elas estão empatadas. Observe como, neste caso o algoritmo preserva a ordem de construção da tabela.

Esta lista quando aplicada no algoritmo gera a seguinte árvore mínima:

```
1,7   1,2   4,7   2,6   2,3   5,6   1,8;
com um custo mínimo de 108
```

☞ Para você fazer

1.

	1	2	3	4	5	6	7	8	9	10
1	0	69	68	89	46	58	49	87	8	11
2	0	0	33	29	33	89	9	63	41	35
3	0	0	0	82	19	36	16	90	39	30
4	0	0	0	0	32	31	70	22	56	82
5	0	0	0	0	0	24	68	18	98	54
6	0	0	0	0	0	0	30	72	57	66
7	0	0	0	0	0	0	0	57	69	30
8	0	0	0	0	0	0	0	0	75	6
9	0	0	0	0	0	0	0	0	0	42
10	0	0	0	0	0	0	0	0	0	0

Pede-se a ÚLTIMA aresta que forma a árvore mínima: de_____a _____,

Pede-se também o CUSTO TOTAL DA ÁRVORE DE COBERTURA MÍNIMA: _____

2.

	1	2	3	4	5	6	7	8	9	10
1	0	87	22	14	62	35	96	29	53	83
2	0	0	10	77	79	23	83	85	51	74
3	0	0	0	87	35	86	36	91	49	13
4	0	0	0	0	74	48	77	93	61	48
5	0	0	0	0	0	26	82	95	85	52
6	0	0	0	0	0	0	88	73	9	83
7	0	0	0	0	0	0	0	69	11	22
8	0	0	0	0	0	0	0	0	50	52
9	0	0	0	0	0	0	0	0	0	11
10	0	0	0	0	0	0	0	0	0	0

Pede-se a ÚLTIMA aresta que forma a árvore mínima: de_____a _____,

Pede-se também o CUSTO TOTAL DA ÁRVORE DE COBERTURA MÍNIMA: _____

PUC/Pr Pontifícia Universidade Católica do Paraná
Curso de Bacharelado em Informatica 30/09/2024 - 10:56:57.9
Cadeira de Estruturas de Dados e Arquivos Prof Dr P Kantek
Árvore de cobertura mínima (alg. Kruskal) VIVX634a V: 3.16
75846 JOAO BONFIN LINO
24PRG502 Exercício : 7 ja vale para P2 de T3 _____ / _____ / _____

Algoritmo de Kruskal: árvore de cobertura mínima

Este problema foi estudado pela primeira vez em 1926 pelo engenheiro Boruvka, quando recebeu a incumbência de eletrificar a Morávia (sudeste da antiga Tchecoslováquia).

Para entender o problema, suponha que você tem uma usina hidrelétrica e um monte de casas espalhadas numa zona rural grande. O objetivo é levar energia elétrica a todas as casas. Ligando todas as casas à usina (ligação em estrela), haverá um grande desperdício de fios e postes.

Outra possibilidade é uma ligação em anel, pela qual são ligadas as casas que estão no perímetro externo da zona, mas permanece a indefinição de como ligar as casas que estão dentro desse perímetro.

Para uma solução ótima (e matematicamente quando se usa o adjetivo ótimo, está se dizendo que é a melhor solução possível), é preciso matematizar o problema, descrevendo-o em termos exatos e passíveis de tratamento matemático. Neste problema, tais termos são o grafo e a matriz de custos. O resultado do problema da ligação é conhecido como árvore de cobertura mínima que nada mais é do que um sub-grafo que conecta todos os vértices e apresenta o menor custo possível.

Dado um grafo não dirigido, cujas arestas comportam custos (distâncias, tempos, volumes, atrasos, postos de pedágio...), este algoritmo gera uma árvore (não há ciclos) que conecta todos os vértices pelo custo mínimo.

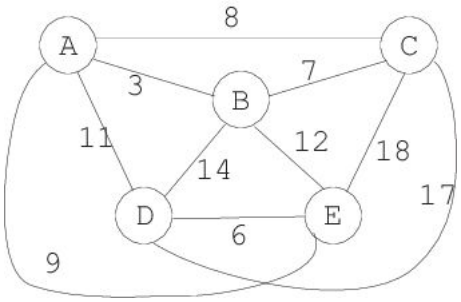
É uma aplicação de "algoritmo guloso"

Suponhamos que o grafo é apresentado através da (diagonal superior da) sua matriz de custos. Por exemplo, seja um grafo de 5 vértices:

Cuja matriz de custos é

	A	B	C	D	E
A	-	3	8	11	9
B	-	-	7	14	12
C	-	-	-	17	18
D	-	-	-	-	6
E	-	-	-	-	-

e cuja figura seja



A etapa 1 do algoritmo é a criação de uma lista de arestas e seus custos

A,B,3 A,C,8 A,D,11 A,E,9 B,C,7
B,D,14 B,E,12 C,D,17 C,E,18 D,E,6

A etapa 2 é ordenar esta lista em ordem crescente de custo e fica:

A,B,3 D,E,6 B,C,7 A,C,8 A,E,9
A,D,11 B,E,12 B,D,14 C,D,17 C,E,18

Etapa 3: Construir um conjunto para cada vértice do grafo

```
1: Inicializar a árvore geradora mínima como vazia
2: enquanto houver mais de um conjunto
3:     Selecionar a aresta de menor custo
4:     se as extremidades estão em conjuntos diferentes
5:         junte os conjuntos e acrescente a aresta na árvore
6:     senão
7:         despreze esta aresta
8:     fim{se}
9: fim{enquanto}
```

Etapa 4: o conjunto (único) contém todos os vértices e a árvore está OK!

Acompanhe o exemplo

- Conjuntos: {A}, {B}, {C}, {D}, {E}, Arvore=[] (vazia)
- Menor: A,B,3. Como A e B estão em conjuntos diferentes, junte-os e fica: {A, B}, {C}, {D}, {E}. Arvore=[(AB)].
- Menor: D,E,6. Como D e E... Fica: {A,B}, {C}, {D,E}. Arvore=[(AB), (DE)]
- menor: B,C,7. Como B e C... Fica {A,B,C}, {D,E}. Arvore=[(AB), (DE), (BC)]
- menor: A,C,8. Como A e C já são do mesmo conjunto, a aresta é desprezada
- menor: A,E,9. Como A e E estão em dois conjuntos, fica:
- {A,B,C,D,E}. Arvore=[(AB), (DE), (BC), (AE)] com um custo de 25 e acabou o algoritmo.

Outro exemplo

Acompanhe este exemplo: Seja o grafo 8 x 8:

	1	2	3	4	5	6	7	8
1	0	7	22	38	64	38	2	37
2	0	0	18	46	44	11	17	52
3	0	0	0	56	43	49	33	59
4	0	0	0	0	44	43	7	43
5	0	0	0	0	0	26	36	45
6	0	0	0	0	0	0	60	47
7	0	0	0	0	0	0	0	44
8	0	0	0	0	0	0	0	0

Que gera a seguinte lista de arestas (**já ordenada**):

CS	OR	DE	CS	OR	DE	CS	OR	DE	CS	OR	DE
==	==	==	==	==	==	==	==	==	==	==	==
2	1	7	26	5	6	43	4	6**	47	6	8
7	1	2**	33	3	7	43	4	8**	49	3	6
7	4	7**	36	5	7	44	2	5**	52	2	8
11	2	6	37	1	8	44	4	5**	56	3	4
17	2	7	38	1	4**	44	7	8**	59	3	8
18	2	3	38	1	6**	45	5	8	60	6	7
22	1	3	43	3	5**	46	2	4	64	1	5

Observação importante: Note as arestas marcadas com **. Elas estão empatadas. Observe como, neste caso o algoritmo preserva a ordem de construção da tabela.

Esta lista quando aplicada no algoritmo gera a seguinte árvore mínima:

```
1,7   1,2   4,7   2,6   2,3   5,6   1,8;
com um custo mínimo de 108
```

☞ Para você fazer

1.

	1	2	3	4	5	6	7	8	9	10
1	0	18	67	93	78	60	52	61	5	65
2	0	0	98	57	45	59	46	24	32	88
3	0	0	0	19	57	9	68	20	15	57
4	0	0	0	0	4	14	23	93	83	9
5	0	0	0	0	0	80	77	10	88	34
6	0	0	0	0	0	0	99	40	69	61
7	0	0	0	0	0	0	0	5	87	3
8	0	0	0	0	0	0	0	0	57	70
9	0	0	0	0	0	0	0	0	0	37
10	0	0	0	0	0	0	0	0	0	0

Pede-se a ÚLTIMA aresta que forma a árvore mínima: de_____a _____,

Pede-se também o CUSTO TOTAL DA ÁRVORE DE COBERTURA MÍNIMA: _____

2.

	1	2	3	4	5	6	7	8	9	10
1	0	87	95	94	96	72	55	14	36	42
2	0	0	53	52	1	76	43	95	73	35
3	0	0	0	55	90	92	58	41	86	50
4	0	0	0	0	54	1	67	46	97	58
5	0	0	0	0	0	5	98	94	25	81
6	0	0	0	0	0	0	91	7	82	52
7	0	0	0	0	0	0	0	51	95	95
8	0	0	0	0	0	0	0	0	67	48
9	0	0	0	0	0	0	0	0	0	62
10	0	0	0	0	0	0	0	0	0	0

Pede-se a ÚLTIMA aresta que forma a árvore mínima: de_____a _____,

Pede-se também o CUSTO TOTAL DA ÁRVORE DE COBERTURA MÍNIMA: _____

PUC/Pr Pontifícia Universidade Católica do Paraná
Curso de Bacharelado em Informatica 30/09/2024 - 10:56:57.9
Cadeira de Estruturas de Dados e Arquivos Prof Dr P Kantek
Árvore de cobertura mínima (alg. Kruskal) VIVX634a V: 3.16
75853 JOAO LUCA FERNANDES DE OLIVE
24PRG502 Exercício : 8 ja vale para P2 de T3 _____ / _____ / _____

Algoritmo de Kruskal: árvore de cobertura mínima

Este problema foi estudado pela primeira vez em 1926 pelo engenheiro Boruvka, quando recebeu a incumbência de eletrificar a Morávia (sudeste da antiga Tchecoslováquia).

Para entender o problema, suponha que você tem uma usina hidrelétrica e um monte de casas espalhadas numa zona rural grande. O objetivo é levar energia elétrica a todas as casas. Ligando todas as casas à usina (ligação em estrela), haverá um grande desperdício de fios e postes.

Outra possibilidade é uma ligação em anel, pela qual são ligadas as casas que estão no perímetro externo da zona, mas permanece a indefinição de como ligar as casas que estão dentro desse perímetro.

Para uma solução ótima (e matematicamente quando se usa o adjetivo ótimo, está se dizendo que é a melhor solução possível), é preciso matematizar o problema, descrevendo-o em termos exatos e passíveis de tratamento matemático. Neste problema, tais termos são o grafo e a matriz de custos. O resultado do problema da ligação é conhecido como árvore de cobertura mínima que nada mais é do que um sub-grafo que conecta todos os vértices e apresenta o menor custo possível.

Dado um grafo não dirigido, cujas arestas comportam custos (distâncias, tempos, volumes, atrasos, postos de pedágio...), este algoritmo gera uma árvore (não há ciclos) que conecta todos os vértices pelo custo mínimo.

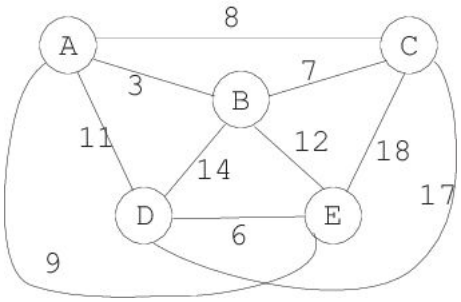
É uma aplicação de "algoritmo guloso"

Suponhamos que o grafo é apresentado através da (diagonal superior da) sua matriz de custos. Por exemplo, seja um grafo de 5 vértices:

Cuja matriz de custos é

	A	B	C	D	E
A	-	3	8	11	9
B	-	-	7	14	12
C	-	-	-	17	18
D	-	-	-	-	6
E	-	-	-	-	-

e cuja figura seja



A etapa 1 do algoritmo é a criação de uma lista de arestas e seus custos

A,B,3 A,C,8 A,D,11 A,E,9 B,C,7
B,D,14 B,E,12 C,D,17 C,E,18 D,E,6

A etapa 2 é ordenar esta lista em ordem crescente de custo e fica:

A,B,3 D,E,6 B,C,7 A,C,8 A,E,9
A,D,11 B,E,12 B,D,14 C,D,17 C,E,18

Etapa 3: Construir um conjunto para cada vértice do grafo

```
1: Inicializar a árvore geradora mínima como vazia
2: enquanto houver mais de um conjunto
3:     Selecionar a aresta de menor custo
4:     se as extremidades estão em conjuntos diferentes
5:         junte os conjuntos e acrescente a aresta na árvore
6:     senão
7:         despreze esta aresta
8:     fim{se}
9: fim{enquanto}
```

Etapa 4: o conjunto (único) contém todos os vértices e a árvore está OK!

Acompanhe o exemplo

- Conjuntos: {A}, {B}, {C}, {D}, {E}, Arvore=[] (vazia)
- Menor: A,B,3. Como A e B estão em conjuntos diferentes, junte-os e fica: {A, B}, {C}, {D}, {E}. Arvore=[(AB)].
- Menor: D,E,6. Como D e E... Fica: {A,B}, {C}, {D,E}. Arvore=[(AB), (DE)]
- menor: B,C,7. Como B e C... Fica {A,B,C}, {D,E}. Arvore=[(AB), (DE), (BC)]
- menor: A,C,8. Como A e C já são do mesmo conjunto, a aresta é desprezada
- menor: A,E,9. Como A e E estão em dois conjuntos, fica:
- {A,B,C,D,E}. Arvore=[(AB), (DE), (BC), (AE)] com um custo de 25 e acabou o algoritmo.

Outro exemplo

Acompanhe este exemplo: Seja o grafo 8 x 8:

	1	2	3	4	5	6	7	8
1	0	7	22	38	64	38	2	37
2	0	0	18	46	44	11	17	52
3	0	0	0	56	43	49	33	59
4	0	0	0	0	44	43	7	43
5	0	0	0	0	0	26	36	45
6	0	0	0	0	0	0	60	47
7	0	0	0	0	0	0	0	44
8	0	0	0	0	0	0	0	0

Que gera a seguinte lista de arestas (**já ordenada**):

CS	OR	DE	CS	OR	DE	CS	OR	DE	CS	OR	DE
==	==	==	==	==	==	==	==	==	==	==	==
2	1	7	26	5	6	43	4	6**	47	6	8
7	1	2**	33	3	7	43	4	8**	49	3	6
7	4	7**	36	5	7	44	2	5**	52	2	8
11	2	6	37	1	8	44	4	5**	56	3	4
17	2	7	38	1	4**	44	7	8**	59	3	8
18	2	3	38	1	6**	45	5	8	60	6	7
22	1	3	43	3	5**	46	2	4	64	1	5

Observação importante: Note as arestas marcadas com **. Elas estão empatadas. Observe como, neste caso o algoritmo preserva a ordem de construção da tabela.

Esta lista quando aplicada no algoritmo gera a seguinte árvore mínima:

```
1,7   1,2   4,7   2,6   2,3   5,6   1,8;
com um custo mínimo de 108
```

☞ Para você fazer

1.

	1	2	3	4	5	6	7	8	9	10
1	0	45	53	39	22	21	81	12	51	38
2	0	0	44	90	8	88	99	90	68	56
3	0	0	0	58	9	24	34	79	63	16
4	0	0	0	0	45	1	39	61	25	11
5	0	0	0	0	0	58	61	30	60	43
6	0	0	0	0	0	0	71	44	31	66
7	0	0	0	0	0	0	0	92	27	99
8	0	0	0	0	0	0	0	0	77	73
9	0	0	0	0	0	0	0	0	0	94
10	0	0	0	0	0	0	0	0	0	0

Pede-se a ÚLTIMA aresta que forma a árvore mínima: de_____a _____,

Pede-se também o CUSTO TOTAL DA ÁRVORE DE COBERTURA MÍNIMA: _____

2.

	1	2	3	4	5	6	7	8	9	10
1	0	50	11	74	99	32	77	14	45	9
2	0	0	5	26	73	57	72	77	21	74
3	0	0	0	13	57	17	38	46	41	37
4	0	0	0	0	64	26	16	74	62	46
5	0	0	0	0	0	8	23	49	65	90
6	0	0	0	0	0	0	75	65	41	99
7	0	0	0	0	0	0	0	34	47	87
8	0	0	0	0	0	0	0	0	15	10
9	0	0	0	0	0	0	0	0	0	38
10	0	0	0	0	0	0	0	0	0	0

Pede-se a ÚLTIMA aresta que forma a árvore mínima: de_____a _____,

Pede-se também o CUSTO TOTAL DA ÁRVORE DE COBERTURA MÍNIMA: _____

PUC/Pr Pontifícia Universidade Católica do Paraná
Curso de Bacharelado em Informatica 30/09/2024 - 10:56:57.9
Cadeira de Estruturas de Dados e Arquivos Prof Dr P Kantek
Árvore de cobertura mínima (alg. Kruskal) VIVX634a V: 3.16
75860 JOAO PEDRO DE BRITO DUARTE
24PRG502 Exercício : 9 ja vale para P2 de T3 _____ / _____ / _____

Algoritmo de Kruskal: árvore de cobertura mínima

Este problema foi estudado pela primeira vez em 1926 pelo engenheiro Boruvka, quando recebeu a incumbência de eletrificar a Morávia (sudeste da antiga Tchecoslováquia).

Para entender o problema, suponha que você tem uma usina hidrelétrica e um monte de casas espalhadas numa zona rural grande. O objetivo é levar energia elétrica a todas as casas. Ligando todas as casas à usina (ligação em estrela), haverá um grande desperdício de fios e postes.

Outra possibilidade é uma ligação em anel, pela qual são ligadas as casas que estão no perímetro externo da zona, mas permanece a indefinição de como ligar as casas que estão dentro desse perímetro.

Para uma solução ótima (e matematicamente quando se usa o adjetivo ótimo, está se dizendo que é a melhor solução possível), é preciso matematizar o problema, descrevendo-o em termos exatos e passíveis de tratamento matemático. Neste problema, tais termos são o grafo e a matriz de custos. O resultado do problema da ligação é conhecido como árvore de cobertura mínima que nada mais é do que um sub-grafo que conecta todos os vértices e apresenta o menor custo possível.

Dado um grafo não dirigido, cujas arestas comportam custos (distâncias, tempos, volumes, atrasos, postos de pedágio...), este algoritmo gera uma árvore (não há ciclos) que conecta todos os vértices pelo custo mínimo.

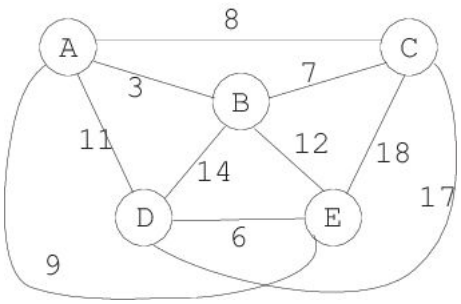
É uma aplicação de "algoritmo guloso"

Suponhamos que o grafo é apresentado através da (diagonal superior da) sua matriz de custos. Por exemplo, seja um grafo de 5 vértices:

Cuja matriz de custos é

	A	B	C	D	E
A	-	3	8	11	9
B	-	-	7	14	12
C	-	-	-	17	18
D	-	-	-	-	6
E	-	-	-	-	-

e cuja figura seja



A etapa 1 do algoritmo é a criação de uma lista de arestas e seus custos

A,B,3 A,C,8 A,D,11 A,E,9 B,C,7
B,D,14 B,E,12 C,D,17 C,E,18 D,E,6

A etapa 2 é ordenar esta lista em ordem crescente de custo e fica:

A,B,3 D,E,6 B,C,7 A,C,8 A,E,9
A,D,11 B,E,12 B,D,14 C,D,17 C,E,18

Etapa 3: Construir um conjunto para cada vértice do grafo

```
1: Inicializar a árvore geradora mínima como vazia
2: enquanto houver mais de um conjunto
3:     Selecionar a aresta de menor custo
4:     se as extremidades estão em conjuntos diferentes
5:         junte os conjuntos e acrescente a aresta na árvore
6:     senão
7:         despreze esta aresta
8:     fim{se}
9: fim{enquanto}
```

Etapa 4: o conjunto (único) contém todos os vértices e a árvore está OK!

Acompanhe o exemplo

- Conjuntos: {A}, {B}, {C}, {D}, {E}, Arvore=[] (vazia)
- Menor: A,B,3. Como A e B estão em conjuntos diferentes, junte-os e fica: {A, B}, {C}, {D}, {E}. Arvore=[(AB)].
- Menor: D,E,6. Como D e E... Fica: {A,B}, {C}, {D,E}. Arvore=[(AB), (DE)]
- menor: B,C,7. Como B e C... Fica {A,B,C}, {D,E}. Arvore=[(AB), (DE), (BC)]
- menor: A,C,8. Como A e C já são do mesmo conjunto, a aresta é desprezada
- menor: A,E,9. Como A e E estão em dois conjuntos, fica:
- {A,B,C,D,E}. Arvore=[(AB), (DE), (BC), (AE)] com um custo de 25 e acabou o algoritmo.

Outro exemplo

Acompanhe este exemplo: Seja o grafo 8 x 8:

	1	2	3	4	5	6	7	8
1	0	7	22	38	64	38	2	37
2	0	0	18	46	44	11	17	52
3	0	0	0	56	43	49	33	59
4	0	0	0	0	44	43	7	43
5	0	0	0	0	0	26	36	45
6	0	0	0	0	0	0	60	47
7	0	0	0	0	0	0	0	44
8	0	0	0	0	0	0	0	0

Que gera a seguinte lista de arestas (**já ordenada**):

CS	OR	DE	CS	OR	DE	CS	OR	DE	CS	OR	DE
==	==	==	==	==	==	==	==	==	==	==	==
2	1	7	26	5	6	43	4	6**	47	6	8
7	1	2**	33	3	7	43	4	8**	49	3	6
7	4	7**	36	5	7	44	2	5**	52	2	8
11	2	6	37	1	8	44	4	5**	56	3	4
17	2	7	38	1	4**	44	7	8**	59	3	8
18	2	3	38	1	6**	45	5	8	60	6	7
22	1	3	43	3	5**	46	2	4	64	1	5

Observação importante: Note as arestas marcadas com **. Elas estão empatadas. Observe como, neste caso o algoritmo preserva a ordem de construção da tabela.

Esta lista quando aplicada no algoritmo gera a seguinte árvore mínima:

```
1,7   1,2   4,7   2,6   2,3   5,6   1,8;
com um custo mínimo de 108
```

☞ Para você fazer

1.

	1	2	3	4	5	6	7	8	9	10
1	0	7	52	34	26	24	82	11	59	24
2	0	0	32	55	3	84	60	97	52	92
3	0	0	0	1	97	93	5	86	55	92
4	0	0	0	0	7	24	17	61	6	10
5	0	0	0	0	0	85	83	13	92	18
6	0	0	0	0	0	0	84	19	5	60
7	0	0	0	0	0	0	0	67	41	71
8	0	0	0	0	0	0	0	0	15	36
9	0	0	0	0	0	0	0	0	0	39
10	0	0	0	0	0	0	0	0	0	0

Pede-se a ÚLTIMA aresta que forma a árvore mínima: de_____a _____,

Pede-se também o CUSTO TOTAL DA ÁRVORE DE COBERTURA MÍNIMA: _____

2.

	1	2	3	4	5	6	7	8	9	10
1	0	33	36	50	8	1	62	26	5	70
2	0	0	7	94	9	38	15	80	55	44
3	0	0	0	61	70	47	96	38	64	3
4	0	0	0	0	16	63	30	48	7	3
5	0	0	0	0	0	25	85	91	24	29
6	0	0	0	0	0	0	59	55	71	37
7	0	0	0	0	0	0	0	94	75	89
8	0	0	0	0	0	0	0	0	37	12
9	0	0	0	0	0	0	0	0	0	26
10	0	0	0	0	0	0	0	0	0	0

Pede-se a ÚLTIMA aresta que forma a árvore mínima: de_____a _____,

Pede-se também o CUSTO TOTAL DA ÁRVORE DE COBERTURA MÍNIMA: _____

PUC/Pr Pontifícia Universidade Católica do Paraná
Curso de Bacharelado em Informatica 30/09/2024 - 10:56:57.9
Cadeira de Estruturas de Dados e Arquivos Prof Dr P Kantek
Árvore de cobertura mínima (alg. Kruskal) VIVX634a V: 3.16
75877 JOAO PEDRO MARIANO SUEKE
24PRG502 Exercício : 10 ja vale para P2 de T3 _____ / _____ / _____

Algoritmo de Kruskal: árvore de cobertura mínima

Este problema foi estudado pela primeira vez em 1926 pelo engenheiro Boruvka, quando recebeu a incumbência de eletrificar a Morávia (sudeste da antiga Tchecoslováquia).

Para entender o problema, suponha que você tem uma usina hidrelétrica e um monte de casas espalhadas numa zona rural grande. O objetivo é levar energia elétrica a todas as casas. Ligando todas as casas à usina (ligação em estrela), haverá um grande desperdício de fios e postes.

Outra possibilidade é uma ligação em anel, pela qual são ligadas as casas que estão no perímetro externo da zona, mas permanece a indefinição de como ligar as casas que estão dentro desse perímetro.

Para uma solução ótima (e matematicamente quando se usa o adjetivo ótimo, está se dizendo que é a melhor solução possível), é preciso matematizar o problema, descrevendo-o em termos exatos e passíveis de tratamento matemático. Neste problema, tais termos são o grafo e a matriz de custos. O resultado do problema da ligação é conhecido como árvore de cobertura mínima que nada mais é do que um sub-grafo que conecta todos os vértices e apresenta o menor custo possível.

Dado um grafo não dirigido, cujas arestas comportam custos (distâncias, tempos, volumes, atrasos, postos de pedágio...), este algoritmo gera uma árvore (não há ciclos) que conecta todos os vértices pelo custo mínimo.

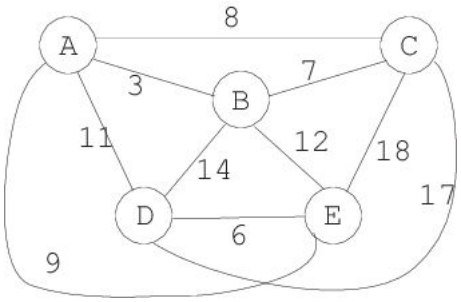
É uma aplicação de "algoritmo guloso"

Suponhamos que o grafo é apresentado através da (diagonal superior da) sua matriz de custos. Por exemplo, seja um grafo de 5 vértices:

Cuja matriz de custos é

	A	B	C	D	E
A	-	3	8	11	9
B	-	-	7	14	12
C	-	-	-	17	18
D	-	-	-	-	6
E	-	-	-	-	-

e cuja figura seja



A etapa 1 do algoritmo é a criação de uma lista de arestas e seus custos

A,B,3 A,C,8 A,D,11 A,E,9 B,C,7
B,D,14 B,E,12 C,D,17 C,E,18 D,E,6

A etapa 2 é ordenar esta lista em ordem crescente de custo e fica:

A,B,3 D,E,6 B,C,7 A,C,8 A,E,9
A,D,11 B,E,12 B,D,14 C,D,17 C,E,18

Etapa 3: Construir um conjunto para cada vértice do grafo

1: Inicializar a árvore geradora mínima como vazia
2: enquanto houver mais de um conjunto
3: Selecionar a aresta de menor custo
4: se as extremidades estão em conjuntos diferentes
5: junte os conjuntos e acrescente a aresta na árvore
6: senão
7: despreze esta aresta
8: fim{se}
9: fim{enquanto}

Etapa 4: o conjunto (único) contém todos os vértices e a árvore está OK!

Acompanhe o exemplo

- Conjuntos: {A}, {B}, {C}, {D}, {E}, Arvore=[] (vazia)
- Menor: A,B,3. Como A e B estão em conjuntos diferentes, junte-os e fica: {A, B}, {C}, {D}, {E}. Arvore=[(AB)].
- Menor: D,E,6. Como D e E... Fica: {A,B}, {C}, {D,E}. Arvore=[(AB), (DE)]
- menor: B,C,7. Como B e C... Fica {A,B,C}, {D,E}. Arvore=[(AB), (DE), (BC)]
- menor: A,C,8. Como A e C já são do mesmo conjunto, a aresta é desprezada
- menor: A,E,9. Como A e E estão em dois conjuntos, fica:
- {A,B,C,D,E}. Arvore=[(AB), (DE), (BC), (AE)] com um custo de 25 e acabou o algoritmo.

Outro exemplo

Acompanhe este exemplo: Seja o grafo 8 x 8:

	1	2	3	4	5	6	7	8
1	0	7	22	38	64	38	2	37
2	0	0	18	46	44	11	17	52
3	0	0	0	56	43	49	33	59
4	0	0	0	0	44	43	7	43
5	0	0	0	0	0	26	36	45
6	0	0	0	0	0	0	60	47
7	0	0	0	0	0	0	0	44
8	0	0	0	0	0	0	0	0

Que gera a seguinte lista de arestas (**já ordenada**):

CS	OR	DE	CS	OR	DE	CS	OR	DE	CS	OR	DE
==	==	==	==	==	==	==	==	==	==	==	==
2	1	7	26	5	6	43	4	6**	47	6	8
7	1	2**	33	3	7	43	4	8**	49	3	6
7	4	7**	36	5	7	44	2	5**	52	2	8
11	2	6	37	1	8	44	4	5**	56	3	4
17	2	7	38	1	4**	44	7	8**	59	3	8
18	2	3	38	1	6**	45	5	8	60	6	7
22	1	3	43	3	5**	46	2	4	64	1	5

Observação importante: Note as arestas marcadas com **. Elas estão empatadas. Observe como, neste caso o algoritmo preserva a ordem de construção da tabela.

Esta lista quando aplicada no algoritmo gera a seguinte árvore mínima:

```
1,7   1,2   4,7   2,6   2,3   5,6   1,8;
com um custo mínimo de 108
```

☞ Para você fazer

1.

	1	2	3	4	5	6	7	8	9	10
1	0	76	65	54	30	20	5	88	62	43
2	0	0	30	11	77	3	19	50	33	60
3	0	0	0	88	21	42	47	2	1	29
4	0	0	0	0	65	32	90	82	40	41
5	0	0	0	0	0	31	82	12	26	46
6	0	0	0	0	0	0	93	92	66	39
7	0	0	0	0	0	0	0	23	54	94
8	0	0	0	0	0	0	0	0	84	8
9	0	0	0	0	0	0	0	0	0	95
10	0	0	0	0	0	0	0	0	0	0

Pede-se a ÚLTIMA aresta que forma a árvore mínima: de_____a _____,

Pede-se também o CUSTO TOTAL DA ÁRVORE DE COBERTURA MÍNIMA: _____

2.

	1	2	3	4	5	6	7	8	9	10
1	0	24	1	57	27	86	92	39	78	93
2	0	0	86	30	79	22	45	10	4	3
3	0	0	0	65	77	25	59	14	72	74
4	0	0	0	0	98	6	45	88	1	12
5	0	0	0	0	0	24	94	85	48	51
6	0	0	0	0	0	0	57	39	88	55
7	0	0	0	0	0	0	0	79	75	20
8	0	0	0	0	0	0	0	0	80	50
9	0	0	0	0	0	0	0	0	0	85
10	0	0	0	0	0	0	0	0	0	0

Pede-se a ÚLTIMA aresta que forma a árvore mínima: de_____a _____,

Pede-se também o CUSTO TOTAL DA ÁRVORE DE COBERTURA MÍNIMA: _____

PUC/Pr Pontifícia Universidade Católica do Paraná
Curso de Bacharelado em Informatica 30/09/2024 - 10:56:57.9
Cadeira de Estruturas de Dados e Arquivos Prof Dr P Kantek
Árvore de cobertura mínima (alg. Kruskal) VIVX634a V: 3.16
75884 JOAO VICTOR GOMES VILELA GER
24PRG502 Exercício : 11 ja vale para P2 de T3 _____ / _____ / _____

Algoritmo de Kruskal: árvore de cobertura mínima

Este problema foi estudado pela primeira vez em 1926 pelo engenheiro Boruvka, quando recebeu a incumbência de eletrificar a Morávia (sudeste da antiga Tchecoslováquia).

Para entender o problema, suponha que você tem uma usina hidrelétrica e um monte de casas espalhadas numa zona rural grande. O objetivo é levar energia elétrica a todas as casas. Ligando todas as casas à usina (ligação em estrela), haverá um grande desperdício de fios e postes.

Outra possibilidade é uma ligação em anel, pela qual são ligadas as casas que estão no perímetro externo da zona, mas permanece a indefinição de como ligar as casas que estão dentro desse perímetro.

Para uma solução ótima (e matematicamente quando se usa o adjetivo ótimo, está se dizendo que é a melhor solução possível), é preciso matematizar o problema, descrevendo-o em termos exatos e passíveis de tratamento matemático. Neste problema, tais termos são o grafo e a matriz de custos. O resultado do problema da ligação é conhecido como árvore de cobertura mínima que nada mais é do que um sub-grafo que conecta todos os vértices e apresenta o menor custo possível.

Dado um grafo não dirigido, cujas arestas comportam custos (distâncias, tempos, volumes, atrasos, postos de pedágio...), este algoritmo gera uma árvore (não há ciclos) que conecta todos os vértices pelo custo mínimo.

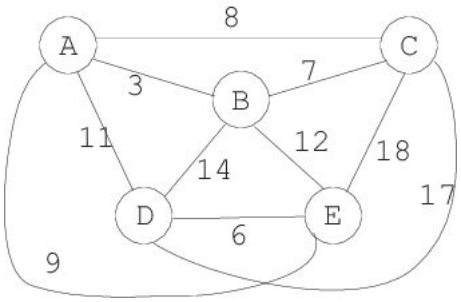
É uma aplicação de "algoritmo guloso"

Suponhamos que o grafo é apresentado através da (diagonal superior da) sua matriz de custos. Por exemplo, seja um grafo de 5 vértices:

Cuja matriz de custos é

	A	B	C	D	E
A	-	3	8	11	9
B	-	-	7	14	12
C	-	-	-	17	18
D	-	-	-	-	6
E	-	-	-	-	-

e cuja figura seja



A etapa 1 do algoritmo é a criação de uma lista de arestas e seus custos

A,B,3 A,C,8 A,D,11 A,E,9 B,C,7
B,D,14 B,E,12 C,D,17 C,E,18 D,E,6

A etapa 2 é ordenar esta lista em ordem crescente de custo e fica:

A,B,3 D,E,6 B,C,7 A,C,8 A,E,9
A,D,11 B,E,12 B,D,14 C,D,17 C,E,18

Etapa 3: Construir um conjunto para cada vértice do grafo

```
1: Inicializar a árvore geradora mínima como vazia
2: enquanto houver mais de um conjunto
3:     Selecionar a aresta de menor custo
4:     se as extremidades estão em conjuntos diferentes
5:         junte os conjuntos e acrescente a aresta na árvore
6:     senão
7:         despreze esta aresta
8:     fim{se}
9: fim{enquanto}
```

Etapa 4: o conjunto (único) contém todos os vértices e a árvore está OK!

Acompanhe o exemplo

- Conjuntos: {A}, {B}, {C}, {D}, {E}, Arvore=[] (vazia)
- Menor: A,B,3. Como A e B estão em conjuntos diferentes, junte-os e fica: {A, B}, {C}, {D}, {E}. Arvore=[(AB)].
- Menor: D,E,6. Como D e E... Fica: {A,B}, {C}, {D,E}. Arvore=[(AB), (DE)]
- menor: B,C,7. Como B e C... Fica {A,B,C}, {D,E}. Arvore=[(AB), (DE), (BC)]
- menor: A,C,8. Como A e C já são do mesmo conjunto, a aresta é desprezada
- menor: A,E,9. Como A e E estão em dois conjuntos, fica:
- {A,B,C,D,E}. Arvore=[(AB), (DE), (BC), (AE)] com um custo de 25 e acabou o algoritmo.

Outro exemplo

Acompanhe este exemplo: Seja o grafo 8 x 8:

	1	2	3	4	5	6	7	8
1	0	7	22	38	64	38	2	37
2	0	0	18	46	44	11	17	52
3	0	0	0	56	43	49	33	59
4	0	0	0	0	44	43	7	43
5	0	0	0	0	0	26	36	45
6	0	0	0	0	0	0	60	47
7	0	0	0	0	0	0	0	44
8	0	0	0	0	0	0	0	0

Que gera a seguinte lista de arestas (**já ordenada**):

CS	OR	DE	CS	OR	DE	CS	OR	DE	CS	OR	DE
==	==	==	==	==	==	==	==	==	==	==	==
2	1	7	26	5	6	43	4	6**	47	6	8
7	1	2**	33	3	7	43	4	8**	49	3	6
7	4	7**	36	5	7	44	2	5**	52	2	8
11	2	6	37	1	8	44	4	5**	56	3	4
17	2	7	38	1	4**	44	7	8**	59	3	8
18	2	3	38	1	6**	45	5	8	60	6	7
22	1	3	43	3	5**	46	2	4	64	1	5

Observação importante: Note as arestas marcadas com **. Elas estão empatadas. Observe como, neste caso o algoritmo preserva a ordem de construção da tabela.

Esta lista quando aplicada no algoritmo gera a seguinte árvore mínima:

```
1,7   1,2   4,7   2,6   2,3   5,6   1,8;
com um custo mínimo de 108
```

☞ Para você fazer

1.

	1	2	3	4	5	6	7	8	9	10
1	0	61	11	33	79	10	87	28	20	16
2	0	0	55	30	34	58	49	59	28	44
3	0	0	0	52	85	95	93	53	53	21
4	0	0	0	0	87	24	52	40	22	15
5	0	0	0	0	0	55	57	85	47	46
6	0	0	0	0	0	0	13	39	51	49
7	0	0	0	0	0	0	0	3	18	78
8	0	0	0	0	0	0	0	0	19	39
9	0	0	0	0	0	0	0	0	0	45
10	0	0	0	0	0	0	0	0	0	0

Pede-se a ÚLTIMA aresta que forma a árvore mínima: de_____a _____,

Pede-se também o CUSTO TOTAL DA ÁRVORE DE COBERTURA MÍNIMA: _____

2.

	1	2	3	4	5	6	7	8	9	10
1	0	55	70	20	8	81	51	88	13	47
2	0	0	92	13	98	70	87	16	72	43
3	0	0	0	52	40	50	7	8	33	8
4	0	0	0	0	77	84	90	93	60	33
5	0	0	0	0	0	37	24	26	58	31
6	0	0	0	0	0	0	60	11	40	69
7	0	0	0	0	0	0	0	12	31	19
8	0	0	0	0	0	0	0	0	16	19
9	0	0	0	0	0	0	0	0	0	13
10	0	0	0	0	0	0	0	0	0	0

Pede-se a ÚLTIMA aresta que forma a árvore mínima: de_____a _____,

Pede-se também o CUSTO TOTAL DA ÁRVORE DE COBERTURA MÍNIMA: _____

PUC/Pr Pontifícia Universidade Católica do Paraná
Curso de Bacharelado em Informatica 30/09/2024 - 10:56:57.9
Cadeira de Estruturas de Dados e Arquivos Prof Dr P Kantek
Árvore de cobertura mínima (alg. Kruskal) VIVX634a V: 3.16
75989 LUCAS ANTOSZCZYSZEN
24PRG502 Exercício : 12 ja vale para P2 de T3 _____ / _____ / _____

Algoritmo de Kruskal: árvore de cobertura mínima

Este problema foi estudado pela primeira vez em 1926 pelo engenheiro Boruvka, quando recebeu a incumbência de eletrificar a Morávia (sudeste da antiga Tchecoslováquia).

Para entender o problema, suponha que você tem uma usina hidrelétrica e um monte de casas espalhadas numa zona rural grande. O objetivo é levar energia elétrica a todas as casas. Ligando todas as casas à usina (ligação em estrela), haverá um grande desperdício de fios e postes.

Outra possibilidade é uma ligação em anel, pela qual são ligadas as casas que estão no perímetro externo da zona, mas permanece a indefinição de como ligar as casas que estão dentro desse perímetro.

Para uma solução ótima (e matematicamente quando se usa o adjetivo ótimo, está se dizendo que é a melhor solução possível), é preciso matematizar o problema, descrevendo-o em termos exatos e passíveis de tratamento matemático. Neste problema, tais termos são o grafo e a matriz de custos. O resultado do problema da ligação é conhecido como árvore de cobertura mínima que nada mais é do que um sub-grafo que conecta todos os vértices e apresenta o menor custo possível.

Dado um grafo não dirigido, cujas arestas comportam custos (distâncias, tempos, volumes, atrasos, postos de pedágio...), este algoritmo gera uma árvore (não há ciclos) que conecta todos os vértices pelo custo mínimo.

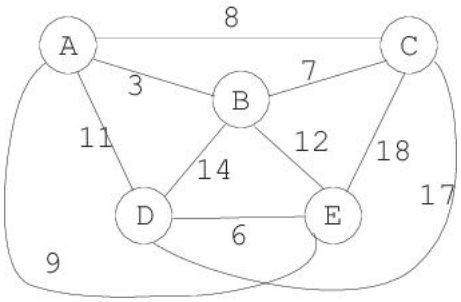
É uma aplicação de "algoritmo guloso"

Suponhamos que o grafo é apresentado através da (diagonal superior da) sua matriz de custos. Por exemplo, seja um grafo de 5 vértices:

Cuja matriz de custos é

	A	B	C	D	E
A	-	3	8	11	9
B	-	-	7	14	12
C	-	-	-	17	18
D	-	-	-	-	6
E	-	-	-	-	-

e cuja figura seja



A etapa 1 do algoritmo é a criação de uma lista de arestas e seus custos

A,B,3 A,C,8 A,D,11 A,E,9 B,C,7
B,D,14 B,E,12 C,D,17 C,E,18 D,E,6

A etapa 2 é ordenar esta lista em ordem crescente de custo e fica:

A,B,3 D,E,6 B,C,7 A,C,8 A,E,9
A,D,11 B,E,12 B,D,14 C,D,17 C,E,18

Etapa 3: Construir um conjunto para cada vértice do grafo

```
1: Inicializar a árvore geradora mínima como vazia
2: enquanto houver mais de um conjunto
3:     Selecionar a aresta de menor custo
4:     se as extremidades estão em conjuntos diferentes
5:         junte os conjuntos e acrescente a aresta na árvore
6:     senão
7:         despreze esta aresta
8:     fim{se}
9: fim{enquanto}
```

Etapa 4: o conjunto (único) contém todos os vértices e a árvore está OK!

Acompanhe o exemplo

- Conjuntos: {A}, {B}, {C}, {D}, {E}, Arvore=[] (vazia)
- Menor: A,B,3. Como A e B estão em conjuntos diferentes, junte-os e fica: {A, B}, {C}, {D}, {E}. Arvore=[(AB)].
- Menor: D,E,6. Como D e E... Fica: {A,B}, {C}, {D,E}. Arvore=[(AB), (DE)]
- menor: B,C,7. Como B e C... Fica {A,B,C}, {D,E}. Arvore=[(AB), (DE), (BC)]
- menor: A,C,8. Como A e C já são do mesmo conjunto, a aresta é desprezada
- menor: A,E,9. Como A e E estão em dois conjuntos, fica:
- {A,B,C,D,E}. Arvore=[(AB), (DE), (BC), (AE)] com um custo de 25 e acabou o algoritmo.

Outro exemplo

Acompanhe este exemplo: Seja o grafo 8 x 8:

	1	2	3	4	5	6	7	8
1	0	7	22	38	64	38	2	37
2	0	0	18	46	44	11	17	52
3	0	0	0	56	43	49	33	59
4	0	0	0	0	44	43	7	43
5	0	0	0	0	0	26	36	45
6	0	0	0	0	0	0	60	47
7	0	0	0	0	0	0	0	44
8	0	0	0	0	0	0	0	0

Que gera a seguinte lista de arestas (**já ordenada**):

CS	OR	DE	CS	OR	DE	CS	OR	DE	CS	OR	DE
==	==	==	==	==	==	==	==	==	==	==	==
2	1	7	26	5	6	43	4	6**	47	6	8
7	1	2**	33	3	7	43	4	8**	49	3	6
7	4	7**	36	5	7	44	2	5**	52	2	8
11	2	6	37	1	8	44	4	5**	56	3	4
17	2	7	38	1	4**	44	7	8**	59	3	8
18	2	3	38	1	6**	45	5	8	60	6	7
22	1	3	43	3	5**	46	2	4	64	1	5

Observação importante: Note as arestas marcadas com **. Elas estão empatadas. Observe como, neste caso o algoritmo preserva a ordem de construção da tabela.

Esta lista quando aplicada no algoritmo gera a seguinte árvore mínima:

```
1,7   1,2   4,7   2,6   2,3   5,6   1,8;
com um custo mínimo de 108
```

☞ Para você fazer

1.

	1	2	3	4	5	6	7	8	9	10
1	0	74	50	48	48	55	34	65	84	11
2	0	0	26	80	70	7	9	62	4	45
3	0	0	0	11	55	51	10	63	61	69
4	0	0	0	0	66	89	91	69	7	61
5	0	0	0	0	0	84	60	39	61	86
6	0	0	0	0	0	0	46	54	19	11
7	0	0	0	0	0	0	0	15	7	81
8	0	0	0	0	0	0	0	0	10	31
9	0	0	0	0	0	0	0	0	0	37
10	0	0	0	0	0	0	0	0	0	0

Pede-se a ÚLTIMA aresta que forma a árvore mínima: de_____a _____,

Pede-se também o CUSTO TOTAL DA ÁRVORE DE COBERTURA MÍNIMA: _____

2.

	1	2	3	4	5	6	7	8	9	10
1	0	43	91	28	40	94	46	4	91	10
2	0	0	72	56	35	12	5	43	21	78
3	0	0	0	92	99	95	53	94	29	79
4	0	0	0	0	80	86	98	3	80	34
5	0	0	0	0	0	17	31	93	25	58
6	0	0	0	0	0	0	5	37	43	6
7	0	0	0	0	0	0	0	45	59	4
8	0	0	0	0	0	0	0	0	1	95
9	0	0	0	0	0	0	0	0	0	2
10	0	0	0	0	0	0	0	0	0	0

Pede-se a ÚLTIMA aresta que forma a árvore mínima: de_____a _____,

Pede-se também o CUSTO TOTAL DA ÁRVORE DE COBERTURA MÍNIMA: _____

PUC/Pr Pontifícia Universidade Católica do Paraná
Curso de Bacharelado em Informatica 30/09/2024 - 10:56:57.9
Cadeira de Estruturas de Dados e Arquivos Prof Dr P Kantek
Árvore de cobertura mínima (alg. Kruskal) VIVX634a V: 3.16
75891 LUIZ FELIPE GUEDES BUCCHERI
24PRG502 Exercício : 13 ja vale para P2 de T3 _____ / _____ / _____

Algoritmo de Kruskal: árvore de cobertura mínima

Este problema foi estudado pela primeira vez em 1926 pelo engenheiro Boruvka, quando recebeu a incumbência de eletrificar a Morávia (sudeste da antiga Tchecoslováquia).

Para entender o problema, suponha que você tem uma usina hidrelétrica e um monte de casas espalhadas numa zona rural grande. O objetivo é levar energia elétrica a todas as casas. Ligando todas as casas à usina (ligação em estrela), haverá um grande desperdício de fios e postes.

Outra possibilidade é uma ligação em anel, pela qual são ligadas as casas que estão no perímetro externo da zona, mas permanece a indefinição de como ligar as casas que estão dentro desse perímetro.

Para uma solução ótima (e matematicamente quando se usa o adjetivo ótimo, está se dizendo que é a melhor solução possível), é preciso matematizar o problema, descrevendo-o em termos exatos e passíveis de tratamento matemático. Neste problema, tais termos são o grafo e a matriz de custos. O resultado do problema da ligação é conhecido como árvore de cobertura mínima que nada mais é do que um sub-grafo que conecta todos os vértices e apresenta o menor custo possível.

Dado um grafo não dirigido, cujas arestas comportam custos (distâncias, tempos, volumes, atrasos, postos de pedágio...), este algoritmo gera uma árvore (não há ciclos) que conecta todos os vértices pelo custo mínimo.

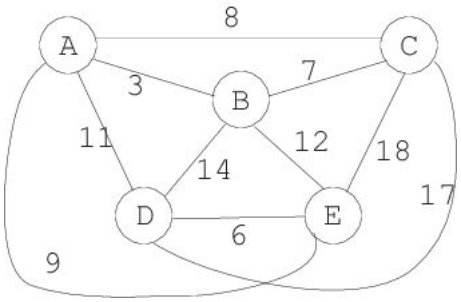
É uma aplicação de "algoritmo guloso"

Suponhamos que o grafo é apresentado através da (diagonal superior da) sua matriz de custos. Por exemplo, seja um grafo de 5 vértices:

Cuja matriz de custos é

	A	B	C	D	E
A	-	3	8	11	9
B	-	-	7	14	12
C	-	-	-	17	18
D	-	-	-	-	6
E	-	-	-	-	-

e cuja figura seja



A etapa 1 do algoritmo é a criação de uma lista de arestas e seus custos

A,B,3 A,C,8 A,D,11 A,E,9 B,C,7
B,D,14 B,E,12 C,D,17 C,E,18 D,E,6

A etapa 2 é ordenar esta lista em ordem crescente de custo e fica:

A,B,3 D,E,6 B,C,7 A,C,8 A,E,9
A,D,11 B,E,12 B,D,14 C,D,17 C,E,18

Etapa 3: Construir um conjunto para cada vértice do grafo

```
1: Inicializar a árvore geradora mínima como vazia
2: enquanto houver mais de um conjunto
3:     Selecionar a aresta de menor custo
4:     se as extremidades estão em conjuntos diferentes
5:         junte os conjuntos e acrescente a aresta na árvore
6:     senão
7:         despreze esta aresta
8:     fim{se}
9: fim{enquanto}
```

Etapa 4: o conjunto (único) contém todos os vértices e a árvore está OK!

Acompanhe o exemplo

- Conjuntos: {A}, {B}, {C}, {D}, {E}, Arvore=[] (vazia)
- Menor: A,B,3. Como A e B estão em conjuntos diferentes, junte-os e fica: {A, B}, {C}, {D}, {E}. Arvore=[(AB)].
- Menor: D,E,6. Como D e E... Fica: {A,B}, {C}, {D,E}. Arvore=[(AB), (DE)]
- menor: B,C,7. Como B e C... Fica {A,B,C}, {D,E}. Arvore=[(AB), (DE), (BC)]
- menor: A,C,8. Como A e C já são do mesmo conjunto, a aresta é desprezada
- menor: A,E,9. Como A e E estão em dois conjuntos, fica:
- {A,B,C,D,E}. Arvore=[(AB), (DE), (BC), (AE)] com um custo de 25 e acabou o algoritmo.

Outro exemplo

Acompanhe este exemplo: Seja o grafo 8 x 8:

	1	2	3	4	5	6	7	8
1	0	7	22	38	64	38	2	37
2	0	0	18	46	44	11	17	52
3	0	0	0	56	43	49	33	59
4	0	0	0	0	44	43	7	43
5	0	0	0	0	0	26	36	45
6	0	0	0	0	0	0	60	47
7	0	0	0	0	0	0	0	44
8	0	0	0	0	0	0	0	0

Que gera a seguinte lista de arestas (**já ordenada**):

CS	OR	DE	CS	OR	DE	CS	OR	DE	CS	OR	DE
==	==	==	==	==	==	==	==	==	==	==	==
2	1	7	26	5	6	43	4	6**	47	6	8
7	1	2**	33	3	7	43	4	8**	49	3	6
7	4	7**	36	5	7	44	2	5**	52	2	8
11	2	6	37	1	8	44	4	5**	56	3	4
17	2	7	38	1	4**	44	7	8**	59	3	8
18	2	3	38	1	6**	45	5	8	60	6	7
22	1	3	43	3	5**	46	2	4	64	1	5

Observação importante: Note as arestas marcadas com **. Elas estão empatadas. Observe como, neste caso o algoritmo preserva a ordem de construção da tabela.

Esta lista quando aplicada no algoritmo gera a seguinte árvore mínima:

```
1,7   1,2   4,7   2,6   2,3   5,6   1,8;
com um custo mínimo de 108
```

☞ Para você fazer

1.

	1	2	3	4	5	6	7	8	9	10
1	0	40	84	21	56	93	4	43	92	12
2	0	0	4	37	93	71	75	20	6	28
3	0	0	0	67	48	6	45	72	48	32
4	0	0	0	0	88	45	27	50	37	85
5	0	0	0	0	0	51	74	13	50	33
6	0	0	0	0	0	0	24	36	48	78
7	0	0	0	0	0	0	0	2	22	39
8	0	0	0	0	0	0	0	0	71	29
9	0	0	0	0	0	0	0	0	0	24
10	0	0	0	0	0	0	0	0	0	0

Pede-se a ÚLTIMA aresta que forma a árvore mínima: de_____a _____,

Pede-se também o CUSTO TOTAL DA ÁRVORE DE COBERTURA MÍNIMA: _____

2.

	1	2	3	4	5	6	7	8	9	10
1	0	24	11	3	39	83	56	68	9	89
2	0	0	71	30	59	22	82	92	39	45
3	0	0	0	98	91	68	87	82	19	63
4	0	0	0	0	16	68	48	22	62	21
5	0	0	0	0	0	35	26	87	59	78
6	0	0	0	0	0	0	29	5	30	19
7	0	0	0	0	0	0	0	73	63	45
8	0	0	0	0	0	0	0	0	81	80
9	0	0	0	0	0	0	0	0	0	5
10	0	0	0	0	0	0	0	0	0	0

Pede-se a ÚLTIMA aresta que forma a árvore mínima: de_____a _____,

Pede-se também o CUSTO TOTAL DA ÁRVORE DE COBERTURA MÍNIMA: _____

PUC/Pr Pontifícia Universidade Católica do Paraná
Curso de Bacharelado em Informatica 30/09/2024 - 10:56:57.9
Cadeira de Estruturas de Dados e Arquivos Prof Dr P Kantek
Árvore de cobertura mínima (alg. Kruskal) VIVX634a V: 3.16
75903 MANUELLA LEAL DE MEIRELLES F
24PRG502 Exercício : 14 ja vale para P2 de T3 _____ / _____ / _____

Algoritmo de Kruskal: árvore de cobertura mínima

Este problema foi estudado pela primeira vez em 1926 pelo engenheiro Boruvka, quando recebeu a incumbência de eletrificar a Morávia (sudeste da antiga Tchecoslováquia).

Para entender o problema, suponha que você tem uma usina hidrelétrica e um monte de casas espalhadas numa zona rural grande. O objetivo é levar energia elétrica a todas as casas. Ligando todas as casas à usina (ligação em estrela), haverá um grande desperdício de fios e postes.

Outra possibilidade é uma ligação em anel, pela qual são ligadas as casas que estão no perímetro externo da zona, mas permanece a indefinição de como ligar as casas que estão dentro desse perímetro.

Para uma solução ótima (e matematicamente quando se usa o adjetivo ótimo, está se dizendo que é a melhor solução possível), é preciso matematizar o problema, descrevendo-o em termos exatos e passíveis de tratamento matemático. Neste problema, tais termos são o grafo e a matriz de custos. O resultado do problema da ligação é conhecido como árvore de cobertura mínima que nada mais é do que um sub-grafo que conecta todos os vértices e apresenta o menor custo possível.

Dado um grafo não dirigido, cujas arestas comportam custos (distâncias, tempos, volumes, atrasos, postos de pedágio...), este algoritmo gera uma árvore (não há ciclos) que conecta todos os vértices pelo custo mínimo.

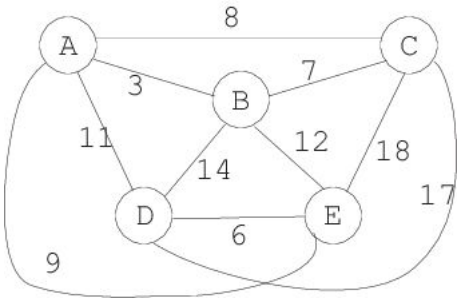
É uma aplicação de "algoritmo guloso"

Suponhamos que o grafo é apresentado através da (diagonal superior da) sua matriz de custos. Por exemplo, seja um grafo de 5 vértices:

Cuja matriz de custos é

	A	B	C	D	E
A	-	3	8	11	9
B	-	-	7	14	12
C	-	-	-	17	18
D	-	-	-	-	6
E	-	-	-	-	-

e cuja figura seja



A etapa 1 do algoritmo é a criação de uma lista de arestas e seus custos

A,B,3 A,C,8 A,D,11 A,E,9 B,C,7
B,D,14 B,E,12 C,D,17 C,E,18 D,E,6

A etapa 2 é ordenar esta lista em ordem crescente de custo e fica:

A,B,3 D,E,6 B,C,7 A,C,8 A,E,9
A,D,11 B,E,12 B,D,14 C,D,17 C,E,18

Etapa 3: Construir um conjunto para cada vértice do grafo

1: Inicializar a árvore geradora mínima como vazia
2: enquanto houver mais de um conjunto
3: Selecionar a aresta de menor custo
4: se as extremidades estão em conjuntos diferentes
5: junte os conjuntos e acrescente a aresta na árvore
6: senão
7: despreze esta aresta
8: fim{se}
9: fim{enquanto}

Etapa 4: o conjunto (único) contém todos os vértices e a árvore está OK!

Acompanhe o exemplo

- Conjuntos: {A}, {B}, {C}, {D}, {E}, Arvore=[] (vazia)
- Menor: A,B,3. Como A e B estão em conjuntos diferentes, junte-os e fica: {A, B}, {C}, {D}, {E}. Arvore=[(AB)].
- Menor: D,E,6. Como D e E... Fica: {A,B}, {C}, {D,E}. Arvore=[(AB), (DE)]
- menor: B,C,7. Como B e C... Fica {A,B,C}, {D,E}. Arvore=[(AB), (DE), (BC)]
- menor: A,C,8. Como A e C já são do mesmo conjunto, a aresta é desprezada
- menor: A,E,9. Como A e E estão em dois conjuntos, fica:
- {A,B,C,D,E}. Arvore=[(AB), (DE), (BC), (AE)] com um custo de 25 e acabou o algoritmo.

Outro exemplo

Acompanhe este exemplo: Seja o grafo 8 x 8:

	1	2	3	4	5	6	7	8
1	0	7	22	38	64	38	2	37
2	0	0	18	46	44	11	17	52
3	0	0	0	56	43	49	33	59
4	0	0	0	0	44	43	7	43
5	0	0	0	0	0	26	36	45
6	0	0	0	0	0	0	60	47
7	0	0	0	0	0	0	0	44
8	0	0	0	0	0	0	0	0

Que gera a seguinte lista de arestas (**já ordenada**):

CS	OR	DE	CS	OR	DE	CS	OR	DE	CS	OR	DE
==	==	==	==	==	==	==	==	==	==	==	==
2	1	7	26	5	6	43	4	6**	47	6	8
7	1	2**	33	3	7	43	4	8**	49	3	6
7	4	7**	36	5	7	44	2	5**	52	2	8
11	2	6	37	1	8	44	4	5**	56	3	4
17	2	7	38	1	4**	44	7	8**	59	3	8
18	2	3	38	1	6**	45	5	8	60	6	7
22	1	3	43	3	5**	46	2	4	64	1	5

Observação importante: Note as arestas marcadas com **. Elas estão empatadas. Observe como, neste caso o algoritmo preserva a ordem de construção da tabela.

Esta lista quando aplicada no algoritmo gera a seguinte árvore mínima:

```
1,7   1,2   4,7   2,6   2,3   5,6   1,8;
com um custo mínimo de 108
```

☞ Para você fazer

1.

	1	2	3	4	5	6	7	8	9	10
1	0	82	6	48	51	6	33	57	53	46
2	0	0	70	14	35	13	65	6	9	56
3	0	0	0	4	61	66	71	95	43	42
4	0	0	0	0	2	7	17	15	60	38
5	0	0	0	0	0	69	62	28	47	31
6	0	0	0	0	0	0	24	22	28	59
7	0	0	0	0	0	0	0	97	13	9
8	0	0	0	0	0	0	0	0	90	10
9	0	0	0	0	0	0	0	0	0	23
10	0	0	0	0	0	0	0	0	0	0

Pede-se a ÚLTIMA aresta que forma a árvore mínima: de_____a _____,

Pede-se também o CUSTO TOTAL DA ÁRVORE DE COBERTURA MÍNIMA: _____

2.

	1	2	3	4	5	6	7	8	9	10
1	0	59	87	10	49	69	67	39	80	73
2	0	0	26	2	96	49	78	44	56	85
3	0	0	0	52	80	41	49	51	79	53
4	0	0	0	0	20	92	2	33	47	15
5	0	0	0	0	0	10	97	31	38	30
6	0	0	0	0	0	0	10	66	96	4
7	0	0	0	0	0	0	0	29	33	33
8	0	0	0	0	0	0	0	0	94	45
9	0	0	0	0	0	0	0	0	0	51
10	0	0	0	0	0	0	0	0	0	0

Pede-se a ÚLTIMA aresta que forma a árvore mínima: de_____a _____,

Pede-se também o CUSTO TOTAL DA ÁRVORE DE COBERTURA MÍNIMA: _____

PUC/Pr Pontifícia Universidade Católica do Paraná
Curso de Bacharelado em Informatica 30/09/2024 - 10:56:57.9
Cadeira de Estruturas de Dados e Arquivos Prof Dr P Kantek
Árvore de cobertura mínima (alg. Kruskal) VIVX634a V: 3.16
75910 MARIA EDUARDA DA SILVA
24PRG502 Exercício : 15 ja vale para P2 de T3 _____ / _____ / _____

Algoritmo de Kruskal: árvore de cobertura mínima

Este problema foi estudado pela primeira vez em 1926 pelo engenheiro Boruvka, quando recebeu a incumbência de eletrificar a Morávia (sudeste da antiga Tchecoslováquia).

Para entender o problema, suponha que você tem uma usina hidrelétrica e um monte de casas espalhadas numa zona rural grande. O objetivo é levar energia elétrica a todas as casas. Ligando todas as casas à usina (ligação em estrela), haverá um grande desperdício de fios e postes.

Outra possibilidade é uma ligação em anel, pela qual são ligadas as casas que estão no perímetro externo da zona, mas permanece a indefinição de como ligar as casas que estão dentro desse perímetro.

Para uma solução ótima (e matematicamente quando se usa o adjetivo ótimo, está se dizendo que é a melhor solução possível), é preciso matematizar o problema, descrevendo-o em termos exatos e passíveis de tratamento matemático. Neste problema, tais termos são o grafo e a matriz de custos. O resultado do problema da ligação é conhecido como árvore de cobertura mínima que nada mais é do que um sub-grafo que conecta todos os vértices e apresenta o menor custo possível.

Dado um grafo não dirigido, cujas arestas comportam custos (distâncias, tempos, volumes, atrasos, postos de pedágio...), este algoritmo gera uma árvore (não há ciclos) que conecta todos os vértices pelo custo mínimo.

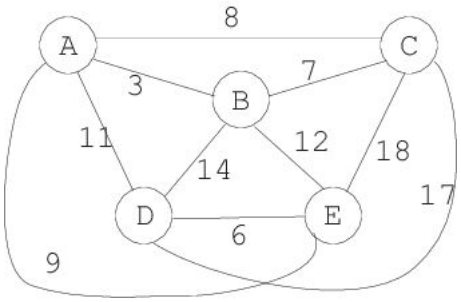
É uma aplicação de "algoritmo guloso"

Suponhamos que o grafo é apresentado através da (diagonal superior da) sua matriz de custos. Por exemplo, seja um grafo de 5 vértices:

Cuja matriz de custos é

	A	B	C	D	E
A	-	3	8	11	9
B	-	-	7	14	12
C	-	-	-	17	18
D	-	-	-	-	6
E	-	-	-	-	-

e cuja figura seja



A etapa 1 do algoritmo é a criação de uma lista de arestas e seus custos

A,B,3 A,C,8 A,D,11 A,E,9 B,C,7
B,D,14 B,E,12 C,D,17 C,E,18 D,E,6

A etapa 2 é ordenar esta lista em ordem crescente de custo e fica:

A,B,3 D,E,6 B,C,7 A,C,8 A,E,9
A,D,11 B,E,12 B,D,14 C,D,17 C,E,18

Etapa 3: Construir um conjunto para cada vértice do grafo

```
1: Inicializar a árvore geradora mínima como vazia
2: enquanto houver mais de um conjunto
3:     Selecionar a aresta de menor custo
4:     se as extremidades estão em conjuntos diferentes
5:         junte os conjuntos e acrescente a aresta na árvore
6:     senão
7:         despreze esta aresta
8:     fim{se}
9: fim{enquanto}
```

Etapa 4: o conjunto (único) contém todos os vértices e a árvore está OK!

Acompanhe o exemplo

- Conjuntos: {A}, {B}, {C}, {D}, {E}, Arvore=[] (vazia)
- Menor: A,B,3. Como A e B estão em conjuntos diferentes, junte-os e fica: {A, B}, {C}, {D}, {E}. Arvore=[(AB)].
- Menor: D,E,6. Como D e E... Fica: {A,B}, {C}, {D,E}. Arvore=[(AB), (DE)]
- menor: B,C,7. Como B e C... Fica {A,B,C}, {D,E}. Arvore=[(AB), (DE), (BC)]
- menor: A,C,8. Como A e C já são do mesmo conjunto, a aresta é desprezada
- menor: A,E,9. Como A e E estão em dois conjuntos, fica:
- {A,B,C,D,E}. Arvore=[(AB), (DE), (BC), (AE)] com um custo de 25 e acabou o algoritmo.

Outro exemplo

Acompanhe este exemplo: Seja o grafo 8 x 8:

	1	2	3	4	5	6	7	8
1	0	7	22	38	64	38	2	37
2	0	0	18	46	44	11	17	52
3	0	0	0	56	43	49	33	59
4	0	0	0	0	44	43	7	43
5	0	0	0	0	0	26	36	45
6	0	0	0	0	0	0	60	47
7	0	0	0	0	0	0	0	44
8	0	0	0	0	0	0	0	0

Que gera a seguinte lista de arestas (**já ordenada**):

CS	OR	DE	CS	OR	DE	CS	OR	DE	CS	OR	DE
==	==	==	==	==	==	==	==	==	==	==	==
2	1	7	26	5	6	43	4	6**	47	6	8
7	1	2**	33	3	7	43	4	8**	49	3	6
7	4	7**	36	5	7	44	2	5**	52	2	8
11	2	6	37	1	8	44	4	5**	56	3	4
17	2	7	38	1	4**	44	7	8**	59	3	8
18	2	3	38	1	6**	45	5	8	60	6	7
22	1	3	43	3	5**	46	2	4	64	1	5

Observação importante: Note as arestas marcadas com **. Elas estão empatadas. Observe como, neste caso o algoritmo preserva a ordem de construção da tabela.

Esta lista quando aplicada no algoritmo gera a seguinte árvore mínima:

```
1,7   1,2   4,7   2,6   2,3   5,6   1,8;
com um custo mínimo de 108
```

☞ Para você fazer

1.

	1	2	3	4	5	6	7	8	9	10
1	0	20	6	18	30	21	25	30	13	29
2	0	0	85	21	35	97	16	97	45	96
3	0	0	0	16	70	78	26	65	47	19
4	0	0	0	0	50	81	54	43	43	84
5	0	0	0	0	0	23	64	89	36	60
6	0	0	0	0	0	0	26	23	76	64
7	0	0	0	0	0	0	0	39	92	30
8	0	0	0	0	0	0	0	0	27	4
9	0	0	0	0	0	0	0	0	0	51
10	0	0	0	0	0	0	0	0	0	0

Pede-se a ÚLTIMA aresta que forma a árvore mínima: de_____a _____,

Pede-se também o CUSTO TOTAL DA ÁRVORE DE COBERTURA MÍNIMA: _____

2.

	1	2	3	4	5	6	7	8	9	10
1	0	46	48	54	1	20	14	9	34	42
2	0	0	87	18	59	12	47	72	18	26
3	0	0	0	81	11	17	74	86	94	66
4	0	0	0	0	74	36	44	34	1	66
5	0	0	0	0	0	36	97	27	50	12
6	0	0	0	0	0	0	57	61	77	34
7	0	0	0	0	0	0	0	84	44	48
8	0	0	0	0	0	0	0	0	38	4
9	0	0	0	0	0	0	0	0	0	95
10	0	0	0	0	0	0	0	0	0	0

Pede-se a ÚLTIMA aresta que forma a árvore mínima: de_____a _____,

Pede-se também o CUSTO TOTAL DA ÁRVORE DE COBERTURA MÍNIMA: _____

PUC/Pr Pontifícia Universidade Católica do Paraná
Curso de Bacharelado em Informatica 30/09/2024 - 10:56:57.9
Cadeira de Estruturas de Dados e Arquivos Prof Dr P Kantek
Árvore de cobertura mínima (alg. Kruskal) VIVX634a V: 3.16
75927 MATHEUS AITA FABRICIO DE CAR
24PRG502 Exercício : 16 ja vale para P2 de T3 _____ / _____ / _____

Algoritmo de Kruskal: árvore de cobertura mínima

Este problema foi estudado pela primeira vez em 1926 pelo engenheiro Boruvka, quando recebeu a incumbência de eletrificar a Morávia (sudeste da antiga Tchecoslováquia).

Para entender o problema, suponha que você tem uma usina hidrelétrica e um monte de casas espalhadas numa zona rural grande. O objetivo é levar energia elétrica a todas as casas. Ligando todas as casas à usina (ligação em estrela), haverá um grande desperdício de fios e postes.

Outra possibilidade é uma ligação em anel, pela qual são ligadas as casas que estão no perímetro externo da zona, mas permanece a indefinição de como ligar as casas que estão dentro desse perímetro.

Para uma solução ótima (e matematicamente quando se usa o adjetivo ótimo, está se dizendo que é a melhor solução possível), é preciso matematizar o problema, descrevendo-o em termos exatos e passíveis de tratamento matemático. Neste problema, tais termos são o grafo e a matriz de custos. O resultado do problema da ligação é conhecido como árvore de cobertura mínima que nada mais é do que um sub-grafo que conecta todos os vértices e apresenta o menor custo possível.

Dado um grafo não dirigido, cujas arestas comportam custos (distâncias, tempos, volumes, atrasos, postos de pedágio...), este algoritmo gera uma árvore (não há ciclos) que conecta todos os vértices pelo custo mínimo.

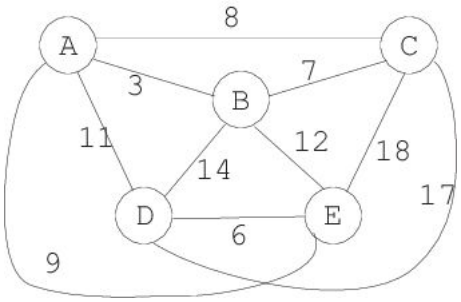
É uma aplicação de "algoritmo guloso"

Suponhamos que o grafo é apresentado através da (diagonal superior da) sua matriz de custos. Por exemplo, seja um grafo de 5 vértices:

Cuja matriz de custos é

	A	B	C	D	E
A	-	3	8	11	9
B	-	-	7	14	12
C	-	-	-	17	18
D	-	-	-	-	6
E	-	-	-	-	-

e cuja figura seja



A etapa 1 do algoritmo é a criação de uma lista de arestas e seus custos

A,B,3 A,C,8 A,D,11 A,E,9 B,C,7
B,D,14 B,E,12 C,D,17 C,E,18 D,E,6

A etapa 2 é ordenar esta lista em ordem crescente de custo e fica:

A,B,3 D,E,6 B,C,7 A,C,8 A,E,9
A,D,11 B,E,12 B,D,14 C,D,17 C,E,18

Etapa 3: Construir um conjunto para cada vértice do grafo

```
1: Inicializar a árvore geradora mínima como vazia
2: enquanto houver mais de um conjunto
3:     Selecionar a aresta de menor custo
4:     se as extremidades estão em conjuntos diferentes
5:         junte os conjuntos e acrescente a aresta na árvore
6:     senão
7:         despreze esta aresta
8:     fim{se}
9: fim{enquanto}
```

Etapa 4: o conjunto (único) contém todos os vértices e a árvore está OK!

Acompanhe o exemplo

- Conjuntos: {A}, {B}, {C}, {D}, {E}, Arvore=[] (vazia)
- Menor: A,B,3. Como A e B estão em conjuntos diferentes, junte-os e fica: {A, B}, {C}, {D}, {E}. Arvore=[(AB)].
- Menor: D,E,6. Como D e E... Fica: {A,B}, {C}, {D,E}. Arvore=[(AB), (DE)]
- menor: B,C,7. Como B e C... Fica {A,B,C}, {D,E}. Arvore=[(AB), (DE), (BC)]
- menor: A,C,8. Como A e C já são do mesmo conjunto, a aresta é desprezada
- menor: A,E,9. Como A e E estão em dois conjuntos, fica:
- {A,B,C,D,E}. Arvore=[(AB), (DE), (BC), (AE)] com um custo de 25 e acabou o algoritmo.

Outro exemplo

Acompanhe este exemplo: Seja o grafo 8 x 8:

	1	2	3	4	5	6	7	8
1	0	7	22	38	64	38	2	37
2	0	0	18	46	44	11	17	52
3	0	0	0	56	43	49	33	59
4	0	0	0	0	44	43	7	43
5	0	0	0	0	0	26	36	45
6	0	0	0	0	0	0	60	47
7	0	0	0	0	0	0	0	44
8	0	0	0	0	0	0	0	0

Que gera a seguinte lista de arestas (**já ordenada**):

CS	OR	DE	CS	OR	DE	CS	OR	DE	CS	OR	DE
==	==	==	==	==	==	==	==	==	==	==	==
2	1	7	26	5	6	43	4	6**	47	6	8
7	1	2**	33	3	7	43	4	8**	49	3	6
7	4	7**	36	5	7	44	2	5**	52	2	8
11	2	6	37	1	8	44	4	5**	56	3	4
17	2	7	38	1	4**	44	7	8**	59	3	8
18	2	3	38	1	6**	45	5	8	60	6	7
22	1	3	43	3	5**	46	2	4	64	1	5

Observação importante: Note as arestas marcadas com **. Elas estão empatadas. Observe como, neste caso o algoritmo preserva a ordem de construção da tabela.

Esta lista quando aplicada no algoritmo gera a seguinte árvore mínima:

```
1,7   1,2   4,7   2,6   2,3   5,6   1,8;
com um custo mínimo de 108
```

☞ Para você fazer

1.

	1	2	3	4	5	6	7	8	9	10
1	0	16	53	3	60	59	48	20	13	87
2	0	0	53	9	90	89	74	56	98	82
3	0	0	0	11	62	19	64	49	5	68
4	0	0	0	0	33	6	24	68	85	25
5	0	0	0	0	0	64	55	69	66	39
6	0	0	0	0	0	0	43	29	51	2
7	0	0	0	0	0	0	0	34	58	79
8	0	0	0	0	0	0	0	0	69	26
9	0	0	0	0	0	0	0	0	0	60
10	0	0	0	0	0	0	0	0	0	0

Pede-se a ÚLTIMA aresta que forma a árvore mínima: de_____a _____,

Pede-se também o CUSTO TOTAL DA ÁRVORE DE COBERTURA MÍNIMA: _____

2.

	1	2	3	4	5	6	7	8	9	10
1	0	30	49	24	1	94	69	47	40	74
2	0	0	12	40	99	3	14	12	95	14
3	0	0	0	73	33	80	51	47	1	52
4	0	0	0	0	67	43	91	23	63	81
5	0	0	0	0	0	96	5	21	39	31
6	0	0	0	0	0	0	70	42	24	65
7	0	0	0	0	0	0	0	58	59	59
8	0	0	0	0	0	0	0	0	7	4
9	0	0	0	0	0	0	0	0	0	71
10	0	0	0	0	0	0	0	0	0	0

Pede-se a ÚLTIMA aresta que forma a árvore mínima: de_____a _____,

Pede-se também o CUSTO TOTAL DA ÁRVORE DE COBERTURA MÍNIMA: _____

PUC/Pr Pontifícia Universidade Católica do Paraná
Curso de Bacharelado em Informatica 30/09/2024 - 10:56:57.9
Cadeira de Estruturas de Dados e Arquivos Prof Dr P Kantek
Árvore de cobertura mínima (alg. Kruskal) VIVX634a V: 3.16
75934 MATHEUS LEANDRO DE BITENCOUR
24PRG502 Exercício : 17 ja vale para P2 de T3 _____ / _____ / _____

Algoritmo de Kruskal: árvore de cobertura mínima

Este problema foi estudado pela primeira vez em 1926 pelo engenheiro Boruvka, quando recebeu a incumbência de eletrificar a Morávia (sudeste da antiga Tchecoslováquia).

Para entender o problema, suponha que você tem uma usina hidrelétrica e um monte de casas espalhadas numa zona rural grande. O objetivo é levar energia elétrica a todas as casas. Ligando todas as casas à usina (ligação em estrela), haverá um grande desperdício de fios e postes.

Outra possibilidade é uma ligação em anel, pela qual são ligadas as casas que estão no perímetro externo da zona, mas permanece a indefinição de como ligar as casas que estão dentro desse perímetro.

Para uma solução ótima (e matematicamente quando se usa o adjetivo ótimo, está se dizendo que é a melhor solução possível), é preciso matematizar o problema, descrevendo-o em termos exatos e passíveis de tratamento matemático. Neste problema, tais termos são o grafo e a matriz de custos. O resultado do problema da ligação é conhecido como árvore de cobertura mínima que nada mais é do que um sub-grafo que conecta todos os vértices e apresenta o menor custo possível.

Dado um grafo não dirigido, cujas arestas comportam custos (distâncias, tempos, volumes, atrasos, postos de pedágio...), este algoritmo gera uma árvore (não há ciclos) que conecta todos os vértices pelo custo mínimo.

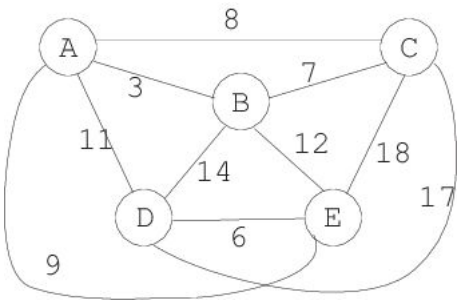
É uma aplicação de "algoritmo guloso"

Suponhamos que o grafo é apresentado através da (diagonal superior da) sua matriz de custos. Por exemplo, seja um grafo de 5 vértices:

Cuja matriz de custos é

	A	B	C	D	E
A	-	3	8	11	9
B	-	-	7	14	12
C	-	-	-	17	18
D	-	-	-	-	6
E	-	-	-	-	-

e cuja figura seja



A etapa 1 do algoritmo é a criação de uma lista de arestas e seus custos

A,B,3 A,C,8 A,D,11 A,E,9 B,C,7
B,D,14 B,E,12 C,D,17 C,E,18 D,E,6

A etapa 2 é ordenar esta lista em ordem crescente de custo e fica:

A,B,3 D,E,6 B,C,7 A,C,8 A,E,9
A,D,11 B,E,12 B,D,14 C,D,17 C,E,18

Etapa 3: Construir um conjunto para cada vértice do grafo

```
1: Inicializar a árvore geradora mínima como vazia
2: enquanto houver mais de um conjunto
3:     Selecionar a aresta de menor custo
4:     se as extremidades estão em conjuntos diferentes
5:         junte os conjuntos e acrescente a aresta na árvore
6:     senão
7:         despreze esta aresta
8:     fim{se}
9: fim{enquanto}
```

Etapa 4: o conjunto (único) contém todos os vértices e a árvore está OK!

Acompanhe o exemplo

- Conjuntos: {A}, {B}, {C}, {D}, {E}, Arvore=[] (vazia)
- Menor: A,B,3. Como A e B estão em conjuntos diferentes, junte-os e fica: {A, B}, {C}, {D}, {E}. Arvore=[(AB)].
- Menor: D,E,6. Como D e E... Fica: {A,B}, {C}, {D,E}. Arvore=[(AB), (DE)]
- menor: B,C,7. Como B e C... Fica {A,B,C}, {D,E}. Arvore=[(AB), (DE), (BC)]
- menor: A,C,8. Como A e C já são do mesmo conjunto, a aresta é desprezada
- menor: A,E,9. Como A e E estão em dois conjuntos, fica:
- {A,B,C,D,E}. Arvore=[(AB), (DE), (BC), (AE)] com um custo de 25 e acabou o algoritmo.

Outro exemplo

Acompanhe este exemplo: Seja o grafo 8 x 8:

	1	2	3	4	5	6	7	8
1	0	7	22	38	64	38	2	37
2	0	0	18	46	44	11	17	52
3	0	0	0	56	43	49	33	59
4	0	0	0	0	44	43	7	43
5	0	0	0	0	0	26	36	45
6	0	0	0	0	0	0	60	47
7	0	0	0	0	0	0	0	44
8	0	0	0	0	0	0	0	0

Que gera a seguinte lista de arestas (**já ordenada**):

CS	OR	DE	CS	OR	DE	CS	OR	DE	CS	OR	DE
==	==	==	==	==	==	==	==	==	==	==	==
2	1	7	26	5	6	43	4	6**	47	6	8
7	1	2**	33	3	7	43	4	8**	49	3	6
7	4	7**	36	5	7	44	2	5**	52	2	8
11	2	6	37	1	8	44	4	5**	56	3	4
17	2	7	38	1	4**	44	7	8**	59	3	8
18	2	3	38	1	6**	45	5	8	60	6	7
22	1	3	43	3	5**	46	2	4	64	1	5

Observação importante: Note as arestas marcadas com **. Elas estão empatadas. Observe como, neste caso o algoritmo preserva a ordem de construção da tabela.

Esta lista quando aplicada no algoritmo gera a seguinte árvore mínima:

```
1,7   1,2   4,7   2,6   2,3   5,6   1,8;
com um custo mínimo de 108
```

☞ Para você fazer

1.

	1	2	3	4	5	6	7	8	9	10
1	0	38	46	42	90	97	40	32	95	71
2	0	0	52	73	49	38	29	40	43	92
3	0	0	0	25	70	52	35	54	46	38
4	0	0	0	0	94	19	76	98	10	9
5	0	0	0	0	0	84	2	41	51	7
6	0	0	0	0	0	0	74	89	62	61
7	0	0	0	0	0	0	0	36	95	71
8	0	0	0	0	0	0	0	0	9	46
9	0	0	0	0	0	0	0	0	0	48
10	0	0	0	0	0	0	0	0	0	0

Pede-se a ÚLTIMA aresta que forma a árvore mínima: de_____a _____,

Pede-se também o CUSTO TOTAL DA ÁRVORE DE COBERTURA MÍNIMA: _____

2.

	1	2	3	4	5	6	7	8	9	10
1	0	47	57	68	79	10	83	76	8	59
2	0	0	9	9	85	23	93	34	20	36
3	0	0	0	57	13	25	35	23	97	24
4	0	0	0	0	46	93	77	55	93	71
5	0	0	0	0	0	9	43	49	13	73
6	0	0	0	0	0	0	38	22	36	70
7	0	0	0	0	0	0	0	69	36	32
8	0	0	0	0	0	0	0	0	4	1
9	0	0	0	0	0	0	0	0	0	46
10	0	0	0	0	0	0	0	0	0	0

Pede-se a ÚLTIMA aresta que forma a árvore mínima: de_____a _____,

Pede-se também o CUSTO TOTAL DA ÁRVORE DE COBERTURA MÍNIMA: _____

PUC/Pr Pontifícia Universidade Católica do Paraná
Curso de Bacharelado em Informatica 30/09/2024 - 10:56:57.9
Cadeira de Estruturas de Dados e Arquivos Prof Dr P Kantek
Árvore de cobertura mínima (alg. Kruskal) VIVX634a V: 3.16
75941 MAYCON TAVARES WOLPE
24PRG502 Exercício : 18 ja vale para P2 de T3 _____ / _____ / _____

Algoritmo de Kruskal: árvore de cobertura mínima

Este problema foi estudado pela primeira vez em 1926 pelo engenheiro Boruvka, quando recebeu a incumbência de eletrificar a Morávia (sudeste da antiga Tchecoslováquia).

Para entender o problema, suponha que você tem uma usina hidrelétrica e um monte de casas espalhadas numa zona rural grande. O objetivo é levar energia elétrica a todas as casas. Ligando todas as casas à usina (ligação em estrela), haverá um grande desperdício de fios e postes.

Outra possibilidade é uma ligação em anel, pela qual são ligadas as casas que estão no perímetro externo da zona, mas permanece a indefinição de como ligar as casas que estão dentro desse perímetro.

Para uma solução ótima (e matematicamente quando se usa o adjetivo ótimo, está se dizendo que é a melhor solução possível), é preciso matematizar o problema, descrevendo-o em termos exatos e passíveis de tratamento matemático. Neste problema, tais termos são o grafo e a matriz de custos. O resultado do problema da ligação é conhecido como árvore de cobertura mínima que nada mais é do que um sub-grafo que conecta todos os vértices e apresenta o menor custo possível.

Dado um grafo não dirigido, cujas arestas comportam custos (distâncias, tempos, volumes, atrasos, postos de pedágio...), este algoritmo gera uma árvore (não há ciclos) que conecta todos os vértices pelo custo mínimo.

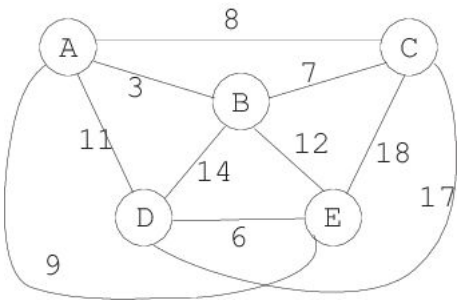
É uma aplicação de "algoritmo guloso"

Suponhamos que o grafo é apresentado através da (diagonal superior da) sua matriz de custos. Por exemplo, seja um grafo de 5 vértices:

Cuja matriz de custos é

	A	B	C	D	E
A	-	3	8	11	9
B	-	-	7	14	12
C	-	-	-	17	18
D	-	-	-	-	6
E	-	-	-	-	-

e cuja figura seja



A etapa 1 do algoritmo é a criação de uma lista de arestas e seus custos

A,B,3 A,C,8 A,D,11 A,E,9 B,C,7
B,D,14 B,E,12 C,D,17 C,E,18 D,E,6

A etapa 2 é ordenar esta lista em ordem crescente de custo e fica:

A,B,3 D,E,6 B,C,7 A,C,8 A,E,9
A,D,11 B,E,12 B,D,14 C,D,17 C,E,18

Etapa 3: Construir um conjunto para cada vértice do grafo

```
1: Inicializar a árvore geradora mínima como vazia
2: enquanto houver mais de um conjunto
3:     Selecionar a aresta de menor custo
4:     se as extremidades estão em conjuntos diferentes
5:         junte os conjuntos e acrescente a aresta na árvore
6:     senão
7:         despreze esta aresta
8:     fim{se}
9: fim{enquanto}
```

Etapa 4: o conjunto (único) contém todos os vértices e a árvore está OK!

Acompanhe o exemplo

- Conjuntos: {A}, {B}, {C}, {D}, {E}, Arvore=[] (vazia)
- Menor: A,B,3. Como A e B estão em conjuntos diferentes, junte-os e fica: {A, B}, {C}, {D}, {E}. Arvore=[(AB)].
- Menor: D,E,6. Como D e E... Fica: {A,B}, {C}, {D,E}. Arvore=[(AB), (DE)]
- menor: B,C,7. Como B e C... Fica {A,B,C}, {D,E}. Arvore=[(AB), (DE), (BC)]
- menor: A,C,8. Como A e C já são do mesmo conjunto, a aresta é desprezada
- menor: A,E,9. Como A e E estão em dois conjuntos, fica:
- {A,B,C,D,E}. Arvore=[(AB), (DE), (BC), (AE)] com um custo de 25 e acabou o algoritmo.

Outro exemplo

Acompanhe este exemplo: Seja o grafo 8 x 8:

	1	2	3	4	5	6	7	8
1	0	7	22	38	64	38	2	37
2	0	0	18	46	44	11	17	52
3	0	0	0	56	43	49	33	59
4	0	0	0	0	44	43	7	43
5	0	0	0	0	0	26	36	45
6	0	0	0	0	0	0	60	47
7	0	0	0	0	0	0	0	44
8	0	0	0	0	0	0	0	0

Que gera a seguinte lista de arestas (**já ordenada**):

CS	OR	DE	CS	OR	DE	CS	OR	DE	CS	OR	DE
==	==	==	==	==	==	==	==	==	==	==	==
2	1	7	26	5	6	43	4	6**	47	6	8
7	1	2**	33	3	7	43	4	8**	49	3	6
7	4	7**	36	5	7	44	2	5**	52	2	8
11	2	6	37	1	8	44	4	5**	56	3	4
17	2	7	38	1	4**	44	7	8**	59	3	8
18	2	3	38	1	6**	45	5	8	60	6	7
22	1	3	43	3	5**	46	2	4	64	1	5

Observação importante: Note as arestas marcadas com **. Elas estão empatadas. Observe como, neste caso o algoritmo preserva a ordem de construção da tabela.

Esta lista quando aplicada no algoritmo gera a seguinte árvore mínima:

```
1,7   1,2   4,7   2,6   2,3   5,6   1,8;
com um custo mínimo de 108
```

☞ Para você fazer

1.

	1	2	3	4	5	6	7	8	9	10
1	0	21	9	14	56	41	64	48	38	61
2	0	0	1	33	96	29	17	65	11	84
3	0	0	0	21	72	33	4	21	47	35
4	0	0	0	0	72	51	30	50	53	43
5	0	0	0	0	0	9	60	55	16	28
6	0	0	0	0	0	0	90	3	19	42
7	0	0	0	0	0	0	0	63	38	86
8	0	0	0	0	0	0	0	0	44	56
9	0	0	0	0	0	0	0	0	0	39
10	0	0	0	0	0	0	0	0	0	0

Pede-se a ÚLTIMA aresta que forma a árvore mínima: de_____a _____,

Pede-se também o CUSTO TOTAL DA ÁRVORE DE COBERTURA MÍNIMA: _____

2.

	1	2	3	4	5	6	7	8	9	10
1	0	48	67	98	68	92	65	84	20	42
2	0	0	45	57	73	85	50	38	49	4
3	0	0	0	94	17	76	87	58	23	30
4	0	0	0	0	28	70	8	44	23	63
5	0	0	0	0	0	23	82	19	20	88
6	0	0	0	0	0	0	66	77	96	34
7	0	0	0	0	0	0	0	14	52	14
8	0	0	0	0	0	0	0	0	66	58
9	0	0	0	0	0	0	0	0	0	46
10	0	0	0	0	0	0	0	0	0	0

Pede-se a ÚLTIMA aresta que forma a árvore mínima: de_____a _____,

Pede-se também o CUSTO TOTAL DA ÁRVORE DE COBERTURA MÍNIMA: _____

PUC/Pr Pontifícia Universidade Católica do Paraná
Curso de Bacharelado em Informatica 30/09/2024 - 10:56:57.9
Cadeira de Estruturas de Dados e Arquivos Prof Dr P Kantek
Árvore de cobertura mínima (alg. Kruskal) VIVX634a V: 3.16
75958 PEDRO TOLEDO LEAL
24PRG502 Exercício : 19 ja vale para P2 de T3 _____ / _____ / _____

Algoritmo de Kruskal: árvore de cobertura mínima

Este problema foi estudado pela primeira vez em 1926 pelo engenheiro Boruvka, quando recebeu a incumbência de eletrificar a Morávia (sudeste da antiga Tchecoslováquia).

Para entender o problema, suponha que você tem uma usina hidrelétrica e um monte de casas espalhadas numa zona rural grande. O objetivo é levar energia elétrica a todas as casas. Ligando todas as casas à usina (ligação em estrela), haverá um grande desperdício de fios e postes.

Outra possibilidade é uma ligação em anel, pela qual são ligadas as casas que estão no perímetro externo da zona, mas permanece a indefinição de como ligar as casas que estão dentro desse perímetro.

Para uma solução ótima (e matematicamente quando se usa o adjetivo ótimo, está se dizendo que é a melhor solução possível), é preciso matematizar o problema, descrevendo-o em termos exatos e passíveis de tratamento matemático. Neste problema, tais termos são o grafo e a matriz de custos. O resultado do problema da ligação é conhecido como árvore de cobertura mínima que nada mais é do que um sub-grafo que conecta todos os vértices e apresenta o menor custo possível.

Dado um grafo não dirigido, cujas arestas comportam custos (distâncias, tempos, volumes, atrasos, postos de pedágio...), este algoritmo gera uma árvore (não há ciclos) que conecta todos os vértices pelo custo mínimo.

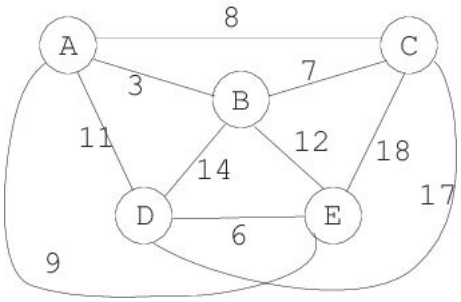
É uma aplicação de "algoritmo guloso"

Suponhamos que o grafo é apresentado através da (diagonal superior da) sua matriz de custos. Por exemplo, seja um grafo de 5 vértices:

Cuja matriz de custos é

	A	B	C	D	E
A	-	3	8	11	9
B	-	-	7	14	12
C	-	-	-	17	18
D	-	-	-	-	6
E	-	-	-	-	-

e cuja figura seja



A etapa 1 do algoritmo é a criação de uma lista de arestas e seus custos

A,B,3 A,C,8 A,D,11 A,E,9 B,C,7
B,D,14 B,E,12 C,D,17 C,E,18 D,E,6

A etapa 2 é ordenar esta lista em ordem crescente de custo e fica:

A,B,3 D,E,6 B,C,7 A,C,8 A,E,9
A,D,11 B,E,12 B,D,14 C,D,17 C,E,18

Etapa 3: Construir um conjunto para cada vértice do grafo

```
1: Inicializar a árvore geradora mínima como vazia
2: enquanto houver mais de um conjunto
3:     Selecionar a aresta de menor custo
4:     se as extremidades estão em conjuntos diferentes
5:         junte os conjuntos e acrescente a aresta na árvore
6:     senão
7:         despreze esta aresta
8:     fim{se}
9: fim{enquanto}
```

Etapa 4: o conjunto (único) contém todos os vértices e a árvore está OK!

Acompanhe o exemplo

- Conjuntos: {A}, {B}, {C}, {D}, {E}, Arvore=[] (vazia)
- Menor: A,B,3. Como A e B estão em conjuntos diferentes, junte-os e fica: {A, B}, {C}, {D}, {E}. Arvore=[(AB)].
- Menor: D,E,6. Como D e E... Fica: {A,B}, {C}, {D,E}. Arvore=[(AB), (DE)]
- menor: B,C,7. Como B e C... Fica {A,B,C}, {D,E}. Arvore=[(AB), (DE), (BC)]
- menor: A,C,8. Como A e C já são do mesmo conjunto, a aresta é desprezada
- menor: A,E,9. Como A e E estão em dois conjuntos, fica:
- {A,B,C,D,E}. Arvore=[(AB), (DE), (BC), (AE)] com um custo de 25 e acabou o algoritmo.

Outro exemplo

Acompanhe este exemplo: Seja o grafo 8 x 8:

	1	2	3	4	5	6	7	8
1	0	7	22	38	64	38	2	37
2	0	0	18	46	44	11	17	52
3	0	0	0	56	43	49	33	59
4	0	0	0	0	44	43	7	43
5	0	0	0	0	0	26	36	45
6	0	0	0	0	0	0	60	47
7	0	0	0	0	0	0	0	44
8	0	0	0	0	0	0	0	0

Que gera a seguinte lista de arestas (**já ordenada**):

CS	OR	DE	CS	OR	DE	CS	OR	DE	CS	OR	DE
==	==	==	==	==	==	==	==	==	==	==	==
2	1	7	26	5	6	43	4	6**	47	6	8
7	1	2**	33	3	7	43	4	8**	49	3	6
7	4	7**	36	5	7	44	2	5**	52	2	8
11	2	6	37	1	8	44	4	5**	56	3	4
17	2	7	38	1	4**	44	7	8**	59	3	8
18	2	3	38	1	6**	45	5	8	60	6	7
22	1	3	43	3	5**	46	2	4	64	1	5

Observação importante: Note as arestas marcadas com **. Elas estão empatadas. Observe como, neste caso o algoritmo preserva a ordem de construção da tabela.

Esta lista quando aplicada no algoritmo gera a seguinte árvore mínima:

```
1,7   1,2   4,7   2,6   2,3   5,6   1,8;
com um custo mínimo de 108
```

☞ Para você fazer

1.

	1	2	3	4	5	6	7	8	9	10
1	0	93	52	72	9	7	37	41	65	94
2	0	0	87	52	36	82	26	72	58	50
3	0	0	0	74	4	6	81	48	57	38
4	0	0	0	0	24	82	72	15	1	84
5	0	0	0	0	0	9	87	49	73	54
6	0	0	0	0	0	0	35	76	45	76
7	0	0	0	0	0	0	0	16	59	63
8	0	0	0	0	0	0	0	0	67	14
9	0	0	0	0	0	0	0	0	0	27
10	0	0	0	0	0	0	0	0	0	0

Pede-se a ÚLTIMA aresta que forma a árvore mínima: de_____a _____,

Pede-se também o CUSTO TOTAL DA ÁRVORE DE COBERTURA MÍNIMA: _____

2.

	1	2	3	4	5	6	7	8	9	10
1	0	52	70	75	14	91	46	21	93	64
2	0	0	11	69	34	43	8	24	20	46
3	0	0	0	63	81	54	16	18	45	97
4	0	0	0	0	63	86	3	84	51	45
5	0	0	0	0	0	35	53	52	5	7
6	0	0	0	0	0	0	93	38	80	22
7	0	0	0	0	0	0	0	92	93	20
8	0	0	0	0	0	0	0	0	62	86
9	0	0	0	0	0	0	0	0	0	88
10	0	0	0	0	0	0	0	0	0	0

Pede-se a ÚLTIMA aresta que forma a árvore mínima: de_____a _____,

Pede-se também o CUSTO TOTAL DA ÁRVORE DE COBERTURA MÍNIMA: _____

PUC/Pr Pontifícia Universidade Católica do Paraná
Curso de Bacharelado em Informatica 30/09/2024 - 10:56:57.9
Cadeira de Estruturas de Dados e Arquivos Prof Dr P Kantek
Árvore de cobertura mínima (alg. Kruskal) VIVX634a V: 3.16
75965 PHELIPE GABRIEL LIMA DA SILV
24PRG502 Exercício : 20 ja vale para P2 de T3 _____ / _____ / _____

Algoritmo de Kruskal: árvore de cobertura mínima

Este problema foi estudado pela primeira vez em 1926 pelo engenheiro Boruvka, quando recebeu a incumbência de eletrificar a Morávia (sudeste da antiga Tchecoslováquia).

Para entender o problema, suponha que você tem uma usina hidrelétrica e um monte de casas espalhadas numa zona rural grande. O objetivo é levar energia elétrica a todas as casas. Ligando todas as casas à usina (ligação em estrela), haverá um grande desperdício de fios e postes.

Outra possibilidade é uma ligação em anel, pela qual são ligadas as casas que estão no perímetro externo da zona, mas permanece a indefinição de como ligar as casas que estão dentro desse perímetro.

Para uma solução ótima (e matematicamente quando se usa o adjetivo ótimo, está se dizendo que é a melhor solução possível), é preciso matematizar o problema, descrevendo-o em termos exatos e passíveis de tratamento matemático. Neste problema, tais termos são o grafo e a matriz de custos. O resultado do problema da ligação é conhecido como árvore de cobertura mínima que nada mais é do que um sub-grafo que conecta todos os vértices e apresenta o menor custo possível.

Dado um grafo não dirigido, cujas arestas comportam custos (distâncias, tempos, volumes, atrasos, postos de pedágio...), este algoritmo gera uma árvore (não há ciclos) que conecta todos os vértices pelo custo mínimo.

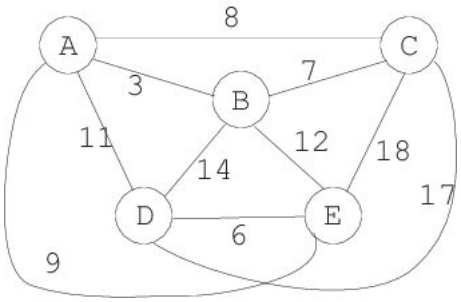
É uma aplicação de "algoritmo guloso"

Suponhamos que o grafo é apresentado através da (diagonal superior da) sua matriz de custos. Por exemplo, seja um grafo de 5 vértices:

Cuja matriz de custos é

	A	B	C	D	E
A	-	3	8	11	9
B	-	-	7	14	12
C	-	-	-	17	18
D	-	-	-	-	6
E	-	-	-	-	-

e cuja figura seja



A etapa 1 do algoritmo é a criação de uma lista de arestas e seus custos

A,B,3 A,C,8 A,D,11 A,E,9 B,C,7
B,D,14 B,E,12 C,D,17 C,E,18 D,E,6

A etapa 2 é ordenar esta lista em ordem crescente de custo e fica:

A,B,3 D,E,6 B,C,7 A,C,8 A,E,9
A,D,11 B,E,12 B,D,14 C,D,17 C,E,18

Etapa 3: Construir um conjunto para cada vértice do grafo

```
1: Inicializar a árvore geradora mínima como vazia
2: enquanto houver mais de um conjunto
3:     Selecionar a aresta de menor custo
4:     se as extremidades estão em conjuntos diferentes
5:         junte os conjuntos e acrescente a aresta na árvore
6:     senão
7:         despreze esta aresta
8:     fim{se}
9: fim{enquanto}
```

Etapa 4: o conjunto (único) contém todos os vértices e a árvore está OK!

Acompanhe o exemplo

- Conjuntos: {A}, {B}, {C}, {D}, {E}, Arvore=[] (vazia)
- Menor: A,B,3. Como A e B estão em conjuntos diferentes, junte-os e fica: {A, B}, {C}, {D}, {E}. Arvore=[(AB)].
- Menor: D,E,6. Como D e E... Fica: {A,B}, {C}, {D,E}. Arvore=[(AB), (DE)]
- menor: B,C,7. Como B e C... Fica {A,B,C}, {D,E}. Arvore=[(AB), (DE), (BC)]
- menor: A,C,8. Como A e C já são do mesmo conjunto, a aresta é desprezada
- menor: A,E,9. Como A e E estão em dois conjuntos, fica:
- {A,B,C,D,E}. Arvore=[(AB), (DE), (BC), (AE)] com um custo de 25 e acabou o algoritmo.

Outro exemplo

Acompanhe este exemplo: Seja o grafo 8 x 8:

	1	2	3	4	5	6	7	8
1	0	7	22	38	64	38	2	37
2	0	0	18	46	44	11	17	52
3	0	0	0	56	43	49	33	59
4	0	0	0	0	44	43	7	43
5	0	0	0	0	0	26	36	45
6	0	0	0	0	0	0	60	47
7	0	0	0	0	0	0	0	44
8	0	0	0	0	0	0	0	0

Que gera a seguinte lista de arestas (**já ordenada**):

CS	OR	DE	CS	OR	DE	CS	OR	DE	CS	OR	DE
==	==	==	==	==	==	==	==	==	==	==	==
2	1	7	26	5	6	43	4	6**	47	6	8
7	1	2**	33	3	7	43	4	8**	49	3	6
7	4	7**	36	5	7	44	2	5**	52	2	8
11	2	6	37	1	8	44	4	5**	56	3	4
17	2	7	38	1	4**	44	7	8**	59	3	8
18	2	3	38	1	6**	45	5	8	60	6	7
22	1	3	43	3	5**	46	2	4	64	1	5

Observação importante: Note as arestas marcadas com **. Elas estão empatadas. Observe como, neste caso o algoritmo preserva a ordem de construção da tabela.

Esta lista quando aplicada no algoritmo gera a seguinte árvore mínima:

```
1,7   1,2   4,7   2,6   2,3   5,6   1,8;
com um custo mínimo de 108
```

☞ Para você fazer

1.

	1	2	3	4	5	6	7	8	9	10
1	0	75	65	17	15	37	44	42	52	36
2	0	0	63	31	32	74	29	55	33	40
3	0	0	0	55	32	94	2	2	8	17
4	0	0	0	0	87	49	16	47	16	75
5	0	0	0	0	0	42	22	88	42	72
6	0	0	0	0	0	0	63	39	9	19
7	0	0	0	0	0	0	0	53	54	86
8	0	0	0	0	0	0	0	0	65	12
9	0	0	0	0	0	0	0	0	0	96
10	0	0	0	0	0	0	0	0	0	0

Pede-se a ÚLTIMA aresta que forma a árvore mínima: de_____a _____,

Pede-se também o CUSTO TOTAL DA ÁRVORE DE COBERTURA MÍNIMA: _____

2.

	1	2	3	4	5	6	7	8	9	10
1	0	86	25	22	85	94	89	92	28	77
2	0	0	5	77	94	72	20	47	13	95
3	0	0	0	65	30	40	80	91	5	51
4	0	0	0	0	57	61	4	86	25	50
5	0	0	0	0	0	66	54	25	21	54
6	0	0	0	0	0	0	98	77	14	58
7	0	0	0	0	0	0	0	42	5	63
8	0	0	0	0	0	0	0	0	36	37
9	0	0	0	0	0	0	0	0	0	66
10	0	0	0	0	0	0	0	0	0	0

Pede-se a ÚLTIMA aresta que forma a árvore mínima: de_____a _____,

Pede-se também o CUSTO TOTAL DA ÁRVORE DE COBERTURA MÍNIMA: _____

PUC/Pr Pontifícia Universidade Católica do Paraná
Curso de Bacharelado em Informatica 30/09/2024 - 10:56:57.9
Cadeira de Estruturas de Dados e Arquivos Prof Dr P Kantek
Árvore de cobertura mínima (alg. Kruskal) VIVX634a V: 3.16
75972 REBECA CABRAL DOS SANTOS
24PRG502 Exercício : 21 ja vale para P2 de T3 _____ / _____ / _____

Algoritmo de Kruskal: árvore de cobertura mínima

Este problema foi estudado pela primeira vez em 1926 pelo engenheiro Boruvka, quando recebeu a incumbência de eletrificar a Morávia (sudeste da antiga Tchecoslováquia).

Para entender o problema, suponha que você tem uma usina hidrelétrica e um monte de casas espalhadas numa zona rural grande. O objetivo é levar energia elétrica a todas as casas. Ligando todas as casas à usina (ligação em estrela), haverá um grande desperdício de fios e postes.

Outra possibilidade é uma ligação em anel, pela qual são ligadas as casas que estão no perímetro externo da zona, mas permanece a indefinição de como ligar as casas que estão dentro desse perímetro.

Para uma solução ótima (e matematicamente quando se usa o adjetivo ótimo, está se dizendo que é a melhor solução possível), é preciso matematizar o problema, descrevendo-o em termos exatos e passíveis de tratamento matemático. Neste problema, tais termos são o grafo e a matriz de custos. O resultado do problema da ligação é conhecido como árvore de cobertura mínima que nada mais é do que um sub-grafo que conecta todos os vértices e apresenta o menor custo possível.

Dado um grafo não dirigido, cujas arestas comportam custos (distâncias, tempos, volumes, atrasos, postos de pedágio...), este algoritmo gera uma árvore (não há ciclos) que conecta todos os vértices pelo custo mínimo.

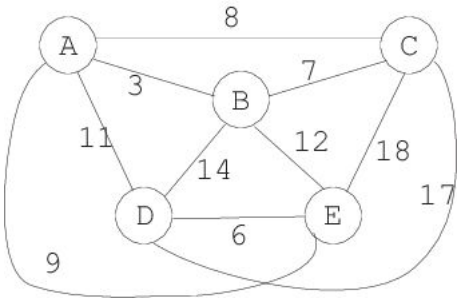
É uma aplicação de "algoritmo guloso"

Suponhamos que o grafo é apresentado através da (diagonal superior da) sua matriz de custos. Por exemplo, seja um grafo de 5 vértices:

Cuja matriz de custos é

	A	B	C	D	E
A	-	3	8	11	9
B	-	-	7	14	12
C	-	-	-	17	18
D	-	-	-	-	6
E	-	-	-	-	-

e cuja figura seja



A etapa 1 do algoritmo é a criação de uma lista de arestas e seus custos

A,B,3	A,C,8	A,D,11	A,E,9	B,C,7
B,D,14	B,E,12	C,D,17	C,E,18	D,E,6

A etapa 2 é ordenar esta lista em ordem crescente de custo e fica:

A,B,3	D,E,6	B,C,7	A,C,8	A,E,9
A,D,11	B,E,12	B,D,14	C,D,17	C,E,18

Etapa 3: Construir um conjunto para cada vértice do grafo

```
1: Inicializar a árvore geradora mínima como vazia
2: enquanto houver mais de um conjunto
3:     Selecionar a aresta de menor custo
4:     se as extremidades estão em conjuntos diferentes
5:         junte os conjuntos e acrescente a aresta na árvore
6:     senão
7:         despreze esta aresta
8:     fim{se}
9: fim{enquanto}
```

Etapa 4: o conjunto (único) contém todos os vértices e a árvore está OK!

Acompanhe o exemplo

- Conjuntos: {A}, {B}, {C}, {D}, {E}, Arvore=[] (vazia)
- Menor: A,B,3. Como A e B estão em conjuntos diferentes, junte-os e fica: {A, B}, {C}, {D}, {E}. Arvore=[(AB)].
- Menor: D,E,6. Como D e E... Fica: {A,B}, {C}, {D,E}. Arvore=[(AB), (DE)]
- menor: B,C,7. Como B e C... Fica {A,B,C}, {D,E}. Arvore=[(AB), (DE), (BC)]
- menor: A,C,8. Como A e C já são do mesmo conjunto, a aresta é desprezada
- menor: A,E,9. Como A e E estão em dois conjuntos, fica:
- {A,B,C,D,E}. Arvore=[(AB), (DE), (BC), (AE)] com um custo de 25 e acabou o algoritmo.

Outro exemplo

Acompanhe este exemplo: Seja o grafo 8 x 8:

	1	2	3	4	5	6	7	8
1	0	7	22	38	64	38	2	37
2	0	0	18	46	44	11	17	52
3	0	0	0	56	43	49	33	59
4	0	0	0	0	44	43	7	43
5	0	0	0	0	0	26	36	45
6	0	0	0	0	0	0	60	47
7	0	0	0	0	0	0	0	44
8	0	0	0	0	0	0	0	0

Que gera a seguinte lista de arestas (**já ordenada**):

CS	OR	DE	CS	OR	DE	CS	OR	DE	CS	OR	DE
==	==	==	==	==	==	==	==	==	==	==	==
2	1	7	26	5	6	43	4	6**	47	6	8
7	1	2**	33	3	7	43	4	8**	49	3	6
7	4	7**	36	5	7	44	2	5**	52	2	8
11	2	6	37	1	8	44	4	5**	56	3	4
17	2	7	38	1	4**	44	7	8**	59	3	8
18	2	3	38	1	6**	45	5	8	60	6	7
22	1	3	43	3	5**	46	2	4	64	1	5

Observação importante: Note as arestas marcadas com **. Elas estão empatadas. Observe como, neste caso o algoritmo preserva a ordem de construção da tabela.

Esta lista quando aplicada no algoritmo gera a seguinte árvore mínima:

```
1,7   1,2   4,7   2,6   2,3   5,6   1,8;
com um custo mínimo de 108
```

☞ Para você fazer

1.

	1	2	3	4	5	6	7	8	9	10
1	0	6	45	44	91	92	40	41	86	77
2	0	0	78	33	4	15	83	55	56	58
3	0	0	0	66	78	22	26	23	37	41
4	0	0	0	0	65	9	68	59	68	58
5	0	0	0	0	0	35	10	98	79	60
6	0	0	0	0	0	0	66	80	49	13
7	0	0	0	0	0	0	0	89	43	34
8	0	0	0	0	0	0	0	0	17	39
9	0	0	0	0	0	0	0	0	0	61
10	0	0	0	0	0	0	0	0	0	0

Pede-se a ÚLTIMA aresta que forma a árvore mínima: de_____a _____,

Pede-se também o CUSTO TOTAL DA ÁRVORE DE COBERTURA MÍNIMA: _____

2.

	1	2	3	4	5	6	7	8	9	10
1	0	47	74	66	60	10	69	41	47	67
2	0	0	97	85	14	41	30	30	53	70
3	0	0	0	49	6	76	37	52	38	35
4	0	0	0	0	33	42	69	67	18	99
5	0	0	0	0	0	56	27	25	67	73
6	0	0	0	0	0	0	31	5	62	67
7	0	0	0	0	0	0	0	77	99	6
8	0	0	0	0	0	0	0	0	77	66
9	0	0	0	0	0	0	0	0	0	45
10	0	0	0	0	0	0	0	0	0	0

Pede-se a ÚLTIMA aresta que forma a árvore mínima: de_____a _____,

Pede-se também o CUSTO TOTAL DA ÁRVORE DE COBERTURA MÍNIMA: _____