PL255 Ex.: 1 _____ / _____ / _____

Árvore QUAD (QUADrante)

O surgimento das árvores QUAD no início dos anos 70, trouxe uma estratégia alternativa na tarefa de armazenar, manipular e analisar imagens computacionais. O nome QUAD vem de QUADrante e a idéia básica é dividir uma imagem em quadrantes de maneira recursiva.

Uma árvore pode ser representada através de uma matriz de cursores. Ganha-se a vantagem do acesso encadeado, sem perder a garantia de que todos os índices (cursores) estarão juntos em uma estrutura sequencial. O preço a pagar por esta facilidade é uma pequena queda na performance.

Combinaremos que a raiz da árvore sempre estará na posição 1 da tabela de cursores, e o endereço de cada filho estará dado pela linha da tabela onde este filho está. Para separar conteúdos de endereços dentro da árvore, estabelecer-se-á: conteúdos=valores negativos e endereços (cursores)=valores positivos.

Dada uma imagem quadrada, a mesma é dividida em quatro quadrantes. Ao mesmo tempo, é criado um nodo contendo 4 filhos na árvore QUAD. Se um determinado quadrante (por exemplo o Q1) é todo ele formado por pixels de mesma cor, esta cor é armazenada no primeiro filho do nodo.

Se outro quadrante (por exemplo o Q3) está formado por pixels de duas cores, o terceiro filho recebe o endereço de outro nodo no qual este quadrante vai ser sub-dividido em quatro novamente.

Acompanhe no exemplo: Seja a imagem 16 x 16 pixels (.=branco; x=preto)

```
                     1 1 1 1 1 1 1 1
   1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8
   - - - - - - - - - - - - - - - - - -
 1-. . . . . . . . . . . . . . . . . .
 2-. . . . . . . . . . . . . . . . . .
 3-. . . . . . . . . . . . . . . . . .
 4-. . . . . . . . . . . . . . . . . .
 5-. . . . . . . . . . . . . . . . . .
 6-. . . . . . . . . . . . . . . . . .
 7-. . . . . . . . . . . . . . . . . .
 8-. . . . . . . . . . x x x x . . . .
 9-. . . . . . . . . . x x x x . . . .
10-. . . . . . . . . . x x x x . . . .
11-. x x x x . . . x x x x x x . . . .
12-. x x x x . . . x x x x . . . . . .
13-. x x x x . . . x x x x . . . . . .
14-. x x x x . . . x x x x . . . . . .
15-. . . . . . . . . . . . . . . . . .
16-. . . . . . . . . . . . . . . . . .
```

Esta imagem geraria a seguinte árvore QUAD:

```
      1-  .  2  7  16
      2-  .  .  3  5
      3-  .  .  .  4
      4-  .  .  x  x
      5-  .  .  6  .
      6-  .  .  x  x
      7-  8  10 12 14
      8-  .  .  9  x
      9-  .  x  .  x
     10-  .  .  11 .
     11-  x  .  x  .
     12-  13 x  .  .
     13-  .  x  .  x
     14-  15 .  .  .
     15-  x  .  x  .
     16-  17 18 20 .
     17-  .  x  x  x
     18-  x  .  19 .
     19-  x  x  .  .
     20-  x  x  .  .
```

Seja agora a imagem de um gato

```
                     1 1 1 1 1 1 1
   1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6
   - - - - - - - - - - - - - - - -
 1-. . . . . . . . . . . . . . . .
 2-. . . . . . . . . . . . . . . .
 3-. . . . x . . . . . . . . . . .
 4-. . x x x . . . . . . . . . . .
 5-. . x x x x . . . . . . . . . .
 6-. . . . x x x . . . . . . . . .
 7-. . . . . x x x x x . . . . . .
 8-. . . . . x x x x x x . . . x x
 9-. . . . . x . x x x x x . . . x
10-. . . . . x . . x x x x . . . x
11-. . . . . x . . x x x x . . x x
12-x x . . x x . x x x x x x x x .
13-. . . . . . . . . . . . . . . .
14-. . . . . . . . . . . . . . . .
15-. . . . . . . . . . . . . . . .
16-. . . . . . . . . . . . . . . .
```

```
 1-2  12 17 25
 2-3  5  7  9
 3-.  .  .  4
 4-.  .  x  x
 5-.  .  6  .
 6-x  .  x  .
 7-.  8  .  .
 8-x  x  .  .
 9-x  10 11 x
10-.  .  x  x
11-.  x  .  x
12-.  .  13 15
13-.  .  x  14
14-.  .  x  .
15-.  .  .  16
```

```
16-.  .  x  x
17-18 20 .  .
18-.  .  19 .
19-.  .  x  x
20-21 22 23 24
21-.  x  .  x
22-.  x  .  .
23-.  x  x  x
24-.  .  .  x
25-x  26 .  .
26-.  27 28 29
27-.  x  .  x
28-.  .  x  x
29-x  x  x  .
```

A função que cria a árvore QUAD pode ser assim escrita

```
 1: função CRIAQUAD (inteiro NUMERO, array IMA)
 2: se linhas(IMA) ≤ 2 então
 3:     AQUAD[NUMERO][1] ← IMA[1][1]
 4:     AQUAD[NUMERO][2] ← IMA[1][2]
 5:     AQUAD[NUMERO][3] ← IMA[2][1]
 6:     AQUAD[NUMERO][4] ← IMA[2][2]
 7: senão
 8:     Q1 ← quadrante(1,IMA)
 9:     Q2 ← quadrante(2,IMA)
10:     Q3 ← quadrante(3,IMA)
11:     Q4 ← quadrante(4,IMA)
12:     se tudoigual(Q1) então
13:         AQUAD[NUMERO;1] ← Q1[1][1]
14:     senão
15:         PROX++
16:         AQUAD[NUMERO][1] ← PROX
17:         CRIAQUAD (PROX, Q1)
18:     fim se
19:     se tudoigual(Q2) então
20:         AQUAD[NUMERO][2] ← Q2[1][1]
21:     senão
22:         PROX++
23:         AQUAD[NUMERO][2] ← PROX
24:         CRIAQUAD (PROX, Q2)
25:     fim se
26:     ... {repete para Q3 e Q4}
27: fim se
28: fim função
```

Para chamar, deve-se criar AQUAD como matriz de 64 por 4 contendo zeros, colocar 1 em PROX, e chamar CRIAQUAD(1, IMAGEM-A-COMPACTAR).

Para reconstituir o desenho, deve-se criar a matriz VOLTA contendo 16x16 brancos, a matriz global AQUAD (que contém a árvore quad) e a variável ONDE, que contém originalmente

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16

A função retiraquadrante recebe a matriz acima e dela retira as linhas e as colunas do quadrante pedido. Por exemplo, para retirar o quadrante 1, ela faz linhas=1^o metade de cima; colunas=1^ometade de baixo. Para quadrante2, linhas=1^ometade de cima; colunas=2^ometade de baixo. Para quadrante3, linhas=2^ometade de cima, colunas=1^ometade de baixo, e assim por diante.

A função que lê a árvore e cria o desenho é a seguinte:

```
 1: função DESENHA (inteiro LINHA, matriz ONDE)
 2: inteiro xx ← AQUAD[LINHA]
 3: se xx[1] = -1 OU xx[1] = -2 então
 4:     VOLTA[retiraquadrante (ONDE, 1)] ← (branco,preto)[abs (xx[1])]
 5: senão
 6:     DESENHA (xx[1], retiraquadrante (ONDE,1))
 7: fim se
 8: se xx[2] = -1 OU xx[2] = -2 então
 9:     VOLTA[retiraquadrante (ONDE, 2)] ← (branco,preto)[abs (xx[2])]
10: senão
11:     DESENHA (xx[2], retiraquadrante (ONDE,2))
12: fim se
13: ... {repete para xx[3] e xx[4] }
14: fim função
```

☞ Para você fazer agora

1. Suponha a imagem

```
                     1 1 1 1 1 1 1
   1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6
   - - - - - - - - - - - - - - - -
 1-. . . . . . . . . . . . . . . .
 2-. . . . . . . . . . . . . . . .
 3-. . . . . . . . . . . . . . . .
 4-. . . . . . . . . . . . . . . .
 5-. . . . . . . . . . . . . . . .
 6-. . . . . . . . . . x x x x . .
 7-. . . . . x x x x . x x x x . .
 8-. . . . . x x x x . x x x x . .
 9-. . . . . x x x x . x x x x . .
10-. . . . . x x x x x x x x . . .
11-. . . . . . . . . . x x x x . . .
12-. . . . . . . . . . x x x x . . .
13-. . . . . . . . . . x x x x . . .
14-. . . . . . . . . . . . . . . .
15-. . . . . . . . . . . . . . . .
16-. . . . . . . . . . . . . . . .
```

Escreva o conteúdo da linha 23 da árvore quad criada pelo algoritmo acima (obs: represente o • por -1 e x por -2).

=_____._____. _____._____.

2. Suponha a árvore quad

```
 1- 2  5  11 14
 2- .  .  .  3
 3- .  .  4  x
 4- .  x  .  x
 5- .  .  6  9
 6- .  7  8  x
 7- .  .  x  x
 8- x  .  x  .
 9- 10 .  x  .
10- .  .  x  x
11- .  12 .  .
```

```
12- 13 x  .  .
13- .  x  .  x
14- 15 18 21 24
15- 16 x  17 x
16- x  .  x  x
17- .  x  .  x
18- 19 .  20 .
19- x  x  x  .
20- x  .  x  .
21- 22 23 .  .
22- .  x  .  .
```

```
23- x  x  .  .
24- 25 .  .  .
25- x  .  .  .
```

Responda qual o conteúdo dos pixels selecionados. O • é -1 e x é -2.

IMAGEM[12][6] = _____

IMAGEM[15][4] = _____