U Positivo - UTFPR - PUCPR - 11/02/2019 - 12:34:01.5 Prof Dr P Kantek (pkantek@up.edu.br) - Recursividade: Inversa e o passeio do cavalo VIVO253a, V: 3.03
1
Exercício : 1 - _____ / _____ / _____

Inversa

Este é o problema em que uma função direta é conhecida (pode ser calculada), enquanto sua inversa não o é. A recursividade pode ser usada para calcular esta inversa por meio se aproximação sucessiva de maneira simples. O algoritmo começa com o valor exato que se quer encontrar e um chute que quando sofresse a ação da função direta deveria gerar o valor exato. Através de modificações desse chute, pode-se chegar rapidamente ao valor esperado. O algoritmo genérico é

```
função ACHAINVERSA (valorreal, valorchutado)
AUX ← funçãodireta(valorchutado)
se 0.0001 > abs(AUX-valorreal)
    devolva valorchutado // é a resposta correta
senão
    devolva ACHAINVERSA (valorreal, NOVOCHUTE(valorchutado))
fim{se}
fimfuncao
```

O segredo deste procedimento está em achar um NOVOCHUTE que converja para a solução. (Aqui, cada caso é um caso e merece apreciação especial). Para simplificar o raciocínio, pode-se estipular que a função sempre é chamada com valorreal = valorchutado.

Seja um exemplo: Uma empresa quer dar um presente de R$ 1.000 a um determinado empregado. Se ela simplesmente adicionar este valor ao salário bruto, o empregado acabará recebendo muito menos, já que os descontos são crescentemente proporcionais aos ganhos. A função direta (dado um bruto achar o líquido) é trivial, mas a inversa não.

Em uma situação hipotética, para um bruto de R$2000, o líquido é de R$ 1258. Se o empregador aumentar o bruto para R$ 3000, o líquido passa a ser de 1723, dando ao empregado apenas (1723-1258=465) 465 reais de presente. A pergunta então é: Quanto deve crescer o bruto para que o líquido seja de 2258 ? (1258+1000).

Definindo:

```
funcao ACHABRU (vreal, vchut)
Z ← CALLIQ (vchut)
se 0.01 > abs(Z-vreal)
    devolva vchut
senão
    devolva ACHABRU (vreal,vchut+(vreal-Z))
fim{se}
fimfunção
```

Chamando ACHABRU (2258,2258) a função acha o líquido para um bruto de 2258 que é 1368. A diferença (1368-2258) é de 890. A nova chamada é ACHABRU (2258,3148). O líquido de 3148 é 1789. A diferença agora é (1789-2258) é 469. A nova chamada é (2258,3617). O líquido de 3617 é 1998. A diferença diminuiu, agora é de 2258-1998=260. A nova chamada é ACHABRU (2258,3877). O líquido de 3877 é 2113... E assim por diante. O valor bruto buscado é de 4202.25, cujo líquido, agora sabemos, é 2258.

O passeio do cavalo

Uma tarefa comum em programação é a busca da solução geral de um problema. Uma estratégia possível (apenas usando computadores) é testar todas as possibilidades mediante o método da tentativa e erro. O caminho mais comum é se decompor a tarefa em sub-tarefas e em geral estas tarefas são muito bem descritas em termos recursivos. Vejamos um exemplo da técnica. Seja um cavalo disposto em um tabuleiro de xadrez de $n \times n = n^2$ casas. (O tabuleiro convencional 8×8 é um caso particular deste mais geral). O cavalo é inicialmente disposto na posição x_0, y_0, e usando os movimentos usuais do xadrez, deve-se verificar se há algum trajeto que leve o cavalo a passar em todas as casas do tabuleiro, entrando uma única vez em cada uma delas. Uma maneira que "quebrar"este problema é buscar solucionar a maneira de localizar qual o próximo movimento possível ou retornar pela impossibilidade de qualquer movimento.

Estruturas de Dados

Usar-se-á uma matriz h de n linhas por n colunas, e nela:

- $h[x, y] = 0$ Indicará que a célula x, y ainda não foi ocupada.
- $h[x, y] = i$ Indicará que a célula x, y foi ocupada no último movimento, onde $1 \leq i \leq n^2$.

A condição tabuleiro não totalmente preenchido pode ser traduzida por $i < n^2$. Se forem criadas duas variáveis u e v a partir das quais se estabelece as novas coordenadas do cavalo, então o predicado movimento aceitável pode ser expresso como $1 \leq u \leq n \wedge 1 \leq v \leq n$ e também $h_{uv} = 0$ ou seja o local de destino do movimento ainda não tenha sido usado. O registro de um movimento válido é então feito usando a atribuição $h_{uv} \leftarrow i$ e o cancelamento deste registro (em caso de chegar-se a um beco sem saída) é pela atribuição $h_{uv} \leftarrow 0$.

Movimentos

Como se sabe, um cavalo localizado na casa central de um tabuleiro 5×5 pode ocupar 8 casas distintas (isto é, pode fazer 8 movimentos distintos. Veja na figura 1.

Algoritmo

e com isso, o algoritmo fica:

```
1: Algoritmo CAVALO
2: inteiro MATRIZ [n] [n]
3: inteiro DLIN [8]
4: inteiro DCOL [8]
5: DLIN ← -2, -1, 1, 2, 2, 1, -1, -2,
6: DCOL ← 1, 2, 2, 1, -1, -2, -2, -1,
7: lógico função ANDACAVALO (int NUMMOV, LCAV, CCAV)
8: inteiro LP, CP
9: inteiro K
10: lógico RESP
11: K ← 1
12: repita
13:     RESP ← FALSO
14:     LP ← LCAV + DLIN [K]
15:     CP ← CCAV + DCOL [K]
16:     se (LP ≥ 1) ∧ (LP ≤ n) ∧ (CP ≥ 1) ∧ (CP ≤ n)
17:         se MATRIZ [LP] [CP] = 0
18:             MATRIZ [LP] [CP] ← NUMMOV
19:             se NUMMOV < (n²)
20:                 RESP ← ANDACAVALO (NUMMOV+1, LP, CP)
21:                 se (RESP = FALSO)
22:                     MATRIZ [LP] [CP] ← 0
23:                 fim{se}
24:             senão
25:                 RESP ← VERDADEIRO
26:             fim{se}
27:         fim{se}
28:     fim{se}
29:     K++
30: até (K > 8) ∨ (RESP = VERDADEIRO))
31: devolva (RESP)
32: fim função
33: MATRIZ ← 0
34: MATRIZ [linha inicial] [coluna inicial] ← 1
35: se (ANDACAVALO (2, linha inicial, coluna inicial))
36:     imprima MATRIZ
37: senão
38:     imprima "Não existe caminho possível"
39: fim{se}
40: fim algoritmo
```

Alguns resultados

Seja um tabuleiro 5×5 inicializado na posição 2,4. Eis uma possível resposta:

25	16	11	6	19
10	5	18	1	12
17	24	15	20	7
4	9	22	13	2
23	14	3	8	21

Este processamento resultou em 86208 movimentos do cavalo.

☞ Para você fazer

Primeiro considere uma modificação no algoritmo acima. Na linha 19, onde se lê $NUMMOV < n^2$ o que equivaleria a testar o valor de i para 64, utilize o valor 56. Ou seja, o algoritmo acaba quando o número 56 for colocado no tabuleiro (e portanto restarem ainda 8 zeros no mesmo). Esta modificação é feita para que o problema possa ser resolvido "à mão".

Para esta instância que você deve resolver, considere a linha inicial do cavalo como sendo a linha 5 e a coluna inicial como sendo a 5 . Para um tabuleiro de 8×8.

Você deve resolver o problema e depois transcrever aqui, o conteúdo da diagonal principal do tabuleiro. Se algum campo não estiver preenchido, responda zero. [1]

- 1 - /

[1] A diagonal que vai de 1,1 até n,n