GabaritoExercíciosProva1

1. Considere dois programas A e B com complexidade $200n^2$ e 22^n, respectivamente. Qual é o mais eficiente?

Valor de n	$200n^2$	22^n
1	200	22
2	800	484
10	20000	2048
12	28800	8192
13	33800	16384
14	39200	32768
15	45000	65536

Para n<= 14 o algoritmo B é mais eficiente
Para n >=15 o algoritmo A é mais eficiente

n	200n^2	22^n
1	200	22
2	800	484
3	1800	10648
4	3200	234256
5	5000	5153632
6	7200	1.13E+08
7	9800	2.49E+09

Para $n \leq 2$ o algoritmo B é mais eficiente, a partir de $n > 2$ o algoritmo A será mais eficiente.

2. Um algoritmo tem complexidade $2n^2$. Num certo computador, num tempo t, o algoritmo resolve um problema de tamanho x. Imagine agora que você tem disponível um computador 30 vezes mais rápido. Que parcela do tempo t precisará para resolver um problema 3 vezes maior?

$T = Comp/VC$

$Comp = 2n^2$

Computador A temos $T = t$ $n = x$ $VC = Vca$

Computador B temos $T = Tb = ??$ $VC = Vcb = 30Vca$ $n = 3x$

Em A $t = 2x^2/Vca$

$Vca = 2x^2 / t$

Em B $Tb = 2(3x)^2 / 30Vca$

$Vca = 18x^2/30Tb$

Igualando os dois Vca obtemos

$2x^2 / t = 18x^2/30Tb$

$Tb = 18t/60$

$Tb = 3t/10$

3. Escreva o pseudocódigo de um algoritmo que troque os valores contido em um arranjo A de n posições pela seguinte política: cada elemento i dentro do arranjo será substituído pela soma de todos os (i-1) elementos mais o elemento i. Por exemplo, dado um arranjo [1; 2; 3; 4; 5] após a aplicação da função teríamos esse arranjo preenchido com os seguintes valores [1; 3; 6; 10; 15].

Solução 1

Algoritmo TroqueValores(A,n)

```
soma = 0
for i=1 to n
    soma = soma + A[i]
    A[i] = soma
```

Solução 2

Algoritmo TroqueValores(A,n)

```
for i = 2 to n
   chave = A[i]
   j= i-1
   A[i] = A[i] + A[j]
```

Solução 3

Troca dos valores

```
For j=1 to A.comprimento
    k = A[j]
    i = j-1
    novoValor = novoValor + k
    A[j] = novoValor
```

Solução 4

```
Algoritmo soma_Ant(A[n])
    Se(n>1)                               C1|1
        For i=1 to n                      C2|n-1
            A[i] = A[i]+A[i-1]            C3|n-1
    retorna A                             C4|1
```

4. Qual invariante de laço esse algoritmo mantém? Usando um invariante de laço, prove que seu algoritmo é correto. Certifique-se de que seu invariante de laço satisfaz as três propriedades necessárias.

A **invariante de laço** para esse pseudocódigo é a seguinte: no início de cada iteração do loop for, indexado por i, a variável soma contem a soma de todos os elementos A[1, 2, ...(i-1)].

Inicialização: Começamos mostrando que a invariante do laço é válida antes da primeira interação do laço, nesse momento a variável i ainda não foi inicializado, portanto trivialmente soma=0.

Manutenção: A cada volta do loop, a variável soma contem a soma de todos os elementos do subvetor A[1.. i-1]

Témino: Quando sair do laço i=n+1 e a variável soma contem a soma de todos os lementos do vetor A[1.. n].

5. Para esse algoritmo forneça os tempos de execução do melhor caso e do pior em notação Θ.

Algoritmo TroqueValores(A,n)	custo	vezes
soma = 0	c_1	1
for i= 1 n	c_2	n + 1
soma = soma + A[i]	c_3	n
A[i] = soma	c_4	n

$T(n) = c_1 + c2(n + 1) + c_3n + c_4n$

$T(n) = (c2 + c_3 + c_4)n + c_1 + 1$

Para o melhor e pior caso $\Theta(n)$