

Universidade Federal de Itajubá

-UNIFEI-

-Itajubá-

-2015-

-Pedro Henrique Dias Scarpioni – 31075

-Prof. Dr. Roberto Claudino – Algoritmos e Grafos

-Bacharelado em Sistemas de Informação

-Instituto de Matemática e Computação – IMC.

-Universidade Federal de Itajubá – UNIFEI.

-Itajubá – MG – Brasil.

Solução Atividade 2 – 28/08

Exercício 1 – Letra A

Função recursiva que computa G.

1 se n = 0

2 então devolve 0

3 senao se n = 1

4 então devolve 1

5 senao se n > 1

6 então devolve 5\* rec(n-1)- 6 \* rec(n-2)

B) O algoritmo acima é de complexidade O, de 2^n, com isso ele cresce exponencialmente, assim sendo, esse algoritmo se mostra ineficiente para entradas muito grandes, isso é para valores altos de n a eficiência deste algoritmo diminui, isto porque muitas operações são repetidas diversas vezes até se chegar a um resultado, que será igual por sinal, assim sendo o algoritmo perde eficiência justamente devido a esse grande numero de operações repetidas.

Exercício 2

Teste de mesa:

: Com um vetor de 5 posições: 1 2 3 4 5

max(A,1,5)

x = 3

a = max(A,1,3)

x = 2

a = max(A,1,2)

x = 1

a = max(A,1,1) → 1

b = max(A,2,2) → 2

R1 = 2

(linha 6)

b = max(A,2,3)

x = 2

a = max(A,2,2) → 2

b = max(A,3,3) → 3

R2 = 3

(linha 6)

b = max(A,4,5)

x = 4

a = max(A,4,4) → 4

b = max(A,5,5) → 5

R3 = 5(linha 6)

R4 = 5 linha(6)

No exemplo acima o vetor em questão possuía 5 elementos, a linha 6, que é a comparação, foi executada 4 vezes, ou seja, para qualquer valor n, a comparação é executada n-1 vezes, assim C(n) = n-1.

Exercício 3

Algoritmo Recursivo

Inserção A,n

If n>1 1

Inserção (a, n-1) t(n-1)

x=a[n] 1

i=n-1 1

while i>0 && a[i]>x n

a[i+1]=a[i] n-1

i=i-1 n-1

a[i+1]=x 1

*T*(*n*)  =  *T*(*n*−1) + 3*n* + 2

Algoritmo sem recursão

Ordenarins A,n

Para j crescendo de 2 ate n faca n

X <- A[j] n-1

I<-j-1 n-1

Enquanto i>0 e a[j]>x faca 2+3+...+n

A[i+1]<-a[i] 1+2+3+....+n-1

I<-i-1 1+2+3+....+n-1

A[i+1]<-x n-1

*T*(*n*)  = 3/2)n2 + 7/2n -4

Comparando os algoritmos recursivo e interativo implementados acima, concluímos que o algoritmo que usa a recursividade é mais eficiente do que o algoritmo que usa a interatividade, se considerarmos o pior caso pra ambos algoritmos, pois a ordem do algoritmo recursivo é O(n) , enquanto o imperativo apresenta ordem de O(n2). Já no melhor caso, concluímos que ambos apresentam ordem O(n), isso se deve pelo fato de que o vetor já vai estar ordenado.