Lista 2

1-) a)

1 int funcao (int n)

2 int g;

3 if(n==0)

4 return 0;

5 else if n==1

6 return 1;

7 else g = (5\*funcao(n-1) – 6\*funcao(n-2));

8 return g;

A função acima computa g.

b) Realizei a complexidade do algoritmo e cheguei à conclusão que no final ele dá uma função exponencial 2n. Pela análise que eu fiz, esse tipo de algoritmo não funciona muito bem se entrada de N for muito grande, pois a medida que N aumenta o fator que está sendo analisado aumenta exponencialmente, diminuindo a eficiência do algoritmo.

2-) vamos ter que calcular a complexidade do algoritmo e entender como ele funciona para justificarmos a resposta

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

Nós concluímos que o algoritmo recursivo é melhor no melhor no pior caso, pois sua eficiência é O n, enquanto o outro algoritmo sem recursão tem eficiência O n2.

Nós concluímos que no melhor caso, os dois algoritmos tem eficiência O n, pois o algoritmo já vai estar ordenado.