UNIVERSIDADE FEDERAL RURAL DE PERNAMBUCO - UFRPE

DEPARTAMENTO DE COMPUTAÇÃO

Algoritmos e Estruturas de Dados

Prof. Filipe Cordeiro

1ª Lista de Exercícios – Complexidade

1. Por muitas vezes damos atenção apenas ao pior caso dos algoritmos, explique o por quê.
2. Que tipo de crescimento melhor caracteriza cada uma dessas funções? (Constante, Linear, Polinomial, Exponencial)
   a) $\left(\frac{3}{2}\right)^n$
   b) 1
   c) $(3/2)n$
   d) $2n^3$
   e) $2n^2$
   f) $3n^2$
   g) 1000
   h) $3n$
3. Classifique as funções de acordo com o crescimento, do crescimento mais lento (na parte de cima) para o crescimento mais rápido (na parte de baixo)
   a. n
   b. n^3
   c. 1
   d. $\left(\frac{3}{2}\right)^n$
   e. n^2
   f. 2^n
4. Classifique as funções de acordo com o crescimento, do crescimento mais lento para o mais rápido.
   a. 4n
   b. $8n^2$
   c. $6n^3$
   d. 64
   e. $n\, log_2 n$
   f. $n \log_6 n$
   g. $\log_2 n$
   h. $\log_8 n$
   i. 8^{2n}
5. Seja um algoritmo com complexidade de tempo $a(n) = n^2 - n + 549$ e B um algoritmo com complexidade de tempo $b(n) = 49n + 49$. Qual algoritmo é melhor?

6. Considere um algoritmo de força bruta para calcular $a^n, onde\ n \in \mathbb{N}$. Pergunta-se:

a. Qual a complexidade desse algoritmo?

b. Construa um algoritmo iterativo que calcule o valor em tempo $O(\log n)$
Para construção do algoritmo, você pode se basear na seguinte fórmula de exponenciação:

$$yx^n = \begin{cases} (yx)(x^2)^{\frac{n-1}{2}}, se\ n \text{ é } ímpar \\ y(x^2)^{\frac{n}{2}}, se\ n \text{ é } par \end{cases}$$

7. Estime a complexidade assintótica de cada um dos algoritmos abaixo:

a.
```
int sum = 0;
for (int n = N; n > 0; n /= 2)
   for (int i = 0; i < n; i++)
      sum++;
```

b.
```
int sum = 0;
for (int i = 1; i < N; i *= 2)
   for(int j = 0; j < i; j++)
      sum++;
```

c.
```
int sum = 0;
for (int i = 1; i < N; i *= 2)
   for (int j = 0; j < N; j++)
      sum++;
```

8. Diga a complexidade assintótica dos seguintes algoritmos:

```
(1)  sum = 0;
     for( i = 0; i < n; i++ )
         sum++;
(2)  sum = 0;
     for( i = 0; i < n; i++ )
         for( j = 0; j < n; j++ )
             sum++;
(3)  sum = 0;
     for( i = 0; i < n; i++ )
         for( j = 0; j < n * n; j++ )
             sum++;
(4)  sum = 0;
     for( i = 0; i < n; i++ )
         for( j = 0; j < i; j++ )
             sum++;
(5)  sum = 0;
     for( i = 0; i < n; i++ )
         for( j = 0; j < i * i; j++ )
             for( k = 0; k < j; k++ )
                 sum++;
```

9.