Lista de exercícios - EDB2

1) Escreva as seguintes funções em notação O:

a) n^3 - 1

b) $n^2 + 2 \log n$

c) $3n^n + 5 \cdot 2^n$

d) $(n - 1)^n + n^{n-1}$

2) Considerando o polinômio quadrático q(n) = $5n^2 + 7n + 3$, mostre que q(n) = O(n^2), determinando constantes $n_0 \in \mathbb{N}$ e c $\in \mathbb{R}_+$ tais que q(n) $\leq cn^2$, para todo n $\geq n_0$.

3)Qual(ais) das seguintes afirmações sobre o crescimento assintótico de funções não é(são) verdadeira(s)?

a) $2n^2 + 3n + 1 = O(n^2)$

b) $n^2 = \Omega(n^3)$

c) Se f(n) = O(g(n)) então g(n) = O(f(n))

d) $\log n^2 = O(\log n)$

e) Se f(n) = O(g(n)) e g(n) = O(h(n)) então f(n) = O(h(n))

f) $2^{n+1} = O(2^n)$

g) $n! = \Theta((n + 1)!)$

4) Considere dois algoritmos A e B que apresentam tempo em $\Theta(n^2)$ e $\Theta(n^3)$, respectivamente, para solucionar um dado problema. Se recursos como memória e tempo de programação não são considerados, é necessariamente verdade que o algoritmo A sempre é preferível ao algoritmo B? Justifique sua resposta.

5) Analise a complexidade local dos algoritmos a seguir e a expresse na notação assintótica O.

```
l ← 0
para i ← 1 até n faça
    para j ← 1 até n² faça
        para k ← 1 até n³ faça
            l ← l + 1
```

```
l ← 0
para i ← 1 até n faça
    para j ← 1 até i faça
        para k ← j até n faça
            l ← l + 1
```

6) Responda as perguntas abaixo com verdadeiro ou falso:

a) 10^{56} . $n^2 \in O(n^2)$?

b) 10^{56} . $n^2 \in O(n^3)$?

c) 10^{56} . $n^2 \in O(n)$?

d) $2^{n+1} \in O(2^n)$?

e) $2^{2n} \in O(2^n)$?

f) $n \in O(n^3)$?

7) Prove ou disprove:

a) $2^{n+1} = O(2^n)$

b) $3^n = O(2^n)$

c) $\log_2 n = O(\log_3 n)$

d) $\log_3 n = O(\log_2 n)$

e) $\log_2 n = O(n)$

f) $100 \log n - 10n + 2n \log n = O(n \log n)$

8) Prove ou disprove as afirmações:

a) $10n = O(n)$

b) $10n^2 = O(n)$

c) $n^2 + 200n + 300 = O(n^2)$

d) $n^2 - 200n - 300 = O(n^2)$

e) $n^2 - 200n - 300 = O(n)$

f) $\frac{3}{2}n^2 + \frac{7}{2}n - 4 = O(n)$

g) $\frac{3}{2}n^2 + \frac{7}{2}\ n - 4 = O(n^2)$

h) $n^3 - 999999n^2 - 1000000 = O(n^2)$

9) Prove ou disprove:

a) Seja $\binom{n}{k}$ o número de combinações de n objetos tomados k a k. Mostre $\binom{n}{2} = \Omega(n^2)$.

b) Prove que 100 log n - 10n + 2n log n está em Ω(n log n).

c) É verdade que 2n + 1 está em Ω(n)?

10) Prove ou disprove:

a) $9999n^2 = \Omega(n^2)$

b) $(\frac{3}{2})n^2 + (\frac{7}{2})n - 4 = \Omega(n^2)$

c) $n^2/1000 - 999n = \Omega(n^2)$

d) $\log_2 n + 1 = \Omega(\log_{10} n)$

11) Responda as perguntas abaixo com verdadeiro ou falso:

a) se $f(n) \in \omega(g(n))$ então $f(n) \in \Omega(g(n))$

b) se $f(n) \in \Omega(g(n))$ então $f(n) \in \omega(g(n))$

c) se $f(n) \in o(g(n))$ então $f(n) \in O(g(n))$

d) se $f(n) \in O(g(n))$ então $f(n) \in o(g(n))$

e) se $f(n) \in \Theta(g(n))$ então $f(n) \in O(g(n))$

f) se $f(n) \in \Theta(g(n))$ então $f(n) \in \Omega(g(n))$

g) se $f(n) \in \Theta(g(n))$ então $f(n) \in o(g(n))$

h) se $f(n) \in \Theta(g(n))$ então $f(n) \in \omega(g(n))$

i) se $f(n) \in \omega(g(n))$ sse $g(n) \in o(f(n))$

12) Considere as seguintes funções de complexidade:

$n^3 + n^2 \log n$, 2^n, $n!$, 3^n, $\log(n) + n^3$, n^2, $10 + n \log(n)$, $20 + \log(n)$, $6\ n \log n + 6$

Ordene-as em ordem de complexidade, separando-as por < quando a da esquerda for de uma classe estritamente menor do que a da direita e por = quando forem da mesma classe.

Por exemplo: f1 < f2 = f3 < f4 = f5 significa que: f1 = O(f2) mas f1 $\neq$ Θ(f2); f2 = Θ(f3); f3 = O(f4) mas f3 $\neq$ Θ(f4); f4 = Θ(f5).

13) Avalie assintoticamente as expressões abaixo usando as notações pertinentes dentre as seguintes: Θ, O e Ω. Caracterize completamente cada caso, usando o menor número de notações possível em cada item. Quer dizer, alguns itens requerem o uso de apenas uma expressão de complexidade, enquanto que outros podem precisar do uso de mais de uma notação. Use o mínimo possível.

a) $O(n) + \Theta(n \log(n))$

b) $O(n \log(n)) + \Theta(n)$

c) $\Omega(n) + \Theta(n^2) + O(n^3)$

d) $\Omega(n) + O(n^2) + \Theta(n^3)$

14) Prove que f(n) é $O(n^2)$, sendo:

a) $f(n) = 2n^2 + 3n + 4$

b) $f(n) = n^2 - 200n - 300$

15) Considere dois algoritmos que executam n operações:

Algoritmo 1: $f_1(n) = 2n^2 + 5n$ operações

Algoritmo 2: $f_2(n) = 500n + 4000$ operações

Qual algoritmo é mais eficiente? Em quais casos?

16) Considere a seguinte sub-rotina:

a) A sub-rotina sempre termina? Explique.

b) Qual a complexidade do algoritmo no pior caso?

c) Qual a complexidade do algoritmo no melhor caso?

d) Quantos passos o algoritmo executa?

```c
int q(int v[], int N, int x) {
    int i;
    int p = -1;
    for (i=0; i < N; i++) {
        if (x == v[i])
            p = i;
    }
    return p;
};
```

e) Qual problema o algortimo se propõe a resolver?

f) Faça uma melhoria em termos de otimização alterando no máximo 4 linhas no código original.

g) Qual a complexidade da melhoria proposta no item f no pior caso?

h) Qual a complexidade da melhoria proposta no item f no melhor caso?

i) Qual a complexidade de caso médio da melhoria proposta no item f?

j) Comente os casos de retorno -1.

k) Comente sobre as disposições e/ou pressuposições do dado de entrada.

17) Considere as seguintes funções de complexidade:

a) N^2

b) N^5

c) $N^2 \log N$

d) N!

e) log N

f) $2*N^2 \log N$

g) $1000*N^5$

h) N log N

i) 99999*N

j) N^3

k) log N + log N

l) 2

m) 2^n

n) 102^{200}

o) 100*N log N

Ordene por ordem de complexidade agrupando funções de classes equivalentes.

18) Considere:

f1 = O(f2) mas f1 $\neq$ Θ(f2); f2 = Θ(f3); f3 = O(f4) mas f3 $\neq$ Θ(f4); f4 = Θ(f5)

Ordene-as em ordem de complexidade, separando-as por "<" ou por "=".

19) Analisar o seguinte algoritmo:

```python
def esta_na_lista(xs, x):
    i = 0
    while i < len(xs):
        if xs[i] == x:
            return True
        i = i + 1
    return False
```

a) Qual é a soma dos custos de todas as linhas?

b) Qual a complexidade assintótica deste algoritmo?

20) Analisar o seguinte algoritmo

a) Qual é a Soma dos custos de todas as linhas?

b) Qual a complexidade assintótica deste algoritmo?

```python
def soma_min_max(xs):
    min = xs[0]
    for x in xs:
        if x < min:
            min = x
    max = xs[0]
    for x in xs:
        if x > xs:
            max = x
    return min + max
```

21) Analisar o seguinte algoritmo

```python
def ordena_insercao(xs):
   for j in range(1, len(xs)):
       chave = xs[j]
       i = j - 1
       while i >= 0 and xs[i] > chave:
           xs[i + 1] = xs[i]
           i = i - 1
       xs[i + 1] = chave
```

a) Qual é a Soma dos custos de todas as linhas?

b) Qual a complexidade no pior caso deste algoritmo?

c) Qual a complexidade no melhor caso deste algoritmo?

22) O que faz e como representar o número de passos do algoritmo:

```c
int mystery(int v[], int n) {
  int count = 0;
  for (int i = 0; i < n; i++)
    for (int j = 0, j < n; j++)
      if (v[i] == v[j])
        count++;
  printf("%i\n", count/2);
}
```

23) Escrever uma função para achar a soma máxima de uma subsequência e expressar seu melhor e pior caso

Dada uma sequência de n valores inteiros (possivelmente negativos) $A = \{a_1, a_2, ..., a_n\}$, encontrar a maior soma de valores contínuos. Ou seja, encontrar

$$\max \sum_{k=i}^{j} a_k$$

Por exemplo, para A = {-2, 11, -4, 13, -5, -2} a resposta é 20 (i.e. a soma de a_2 até a_4)

24) Sobre o Problema de medir eficiência de algoritmos, escreva a respeito do método experimental e do método analítico, exemplifique os métodos e comente os pontos positivos e negativos de cada método.

25) Prove ou disprove:

a) $10^{56}\ n^2 \in \Omega(n^2)$?

b) $10^{56}\ n^2 \in \Omega(n^3)$?

c) $10^{56}\ n^2 \in \Omega(n)$?

d) $2^{n+1} \in \Omega(2^n)$?

e) $2^{2n} \in \Omega(2^n)$?

f) $n \in \Omega(n^3)$?

26) Prove ou disprove:

a) $2^{n+1} = \Omega(2^n)$

b) $3^n = \Omega(2^n)$

c) $\log_2 n = \Omega(\log_3 n)$

d) $\log_3 n = \Omega(\log_2 n)$

e) $\log_2 n = \Omega(n)$

f) $100 \log n - 10n + 2n \log n = \Omega(n \log n)$

27) Prove ou disprove:

a) $10n = \Omega(n)$

b) $10n^2 = \Omega(n)$

c) $n^2 + 200n + 300 = \Omega(n^2)$

d) $n^2 - 200n - 300 = \Omega(n^2)$

e) $n^2 - 200n - 300 = \Omega(n)$

f) $(\frac{3}{2})n^2 + (\frac{7}{2})n - 4 = \Omega(n)$

g) $(\frac{3}{2})n^2 + (\frac{7}{2})n - 4 = \Omega(n^2)$

h) $n^3 - 999999n^2 - 1000000 = \Omega(n^2)$

28) Resolva as questões abaixo:

a) Seja $\binom{n}{k}$ o número de combinações de n objetos tomados k a k. Mostre $\binom{n}{2} = O(n^2)$.

b) Prove que 100 log n - 10n + 2n log n está em O(n log n).

c) É verdade que 2n + 1 está em O(n)?

29) Prove ou disprove:

a) $9999n^2 = \Theta(n^2)$

b) $(3/2)n^2 + (7/2)n - 4 = \Theta(n^2)$

c) $n^2 / 1000 - 999n = \Theta(n^2)$

d) $\log_2 n + 1 = \Theta(\log_{10} n)$

30) Indicar Verdadeiro ou Falso para as sentenças abaixo:

a) se $f(n) \in \omega(g(n))$ então $f(n) \in \Omega(g(n))$

b) se $f(n) \in \Omega(g(n))$ então $f(n) \in \omega(g(n))$

c) se $f(n) \in o(g(n))$ então $f(n) \in O(g(n))$

d) se $f(n) \in O(g(n))$ então $f(n) \in o(g(n))$

e) se $f(n) \in \Theta(g(n))$ então $f(n) \in O(g(n))$

f) se $f(n) \in \Theta(g(n))$ então $f(n) \in \Omega(g(n))$

g) se $f(n) \in \Theta(g(n))$ então $f(n) \in o(g(n))$

h) se $f(n) \in \Theta(g(n))$ então $f(n) \in \omega(g(n))$

i) se $f(n) \in \omega(g(n))$ sse $g(n) \in o(f(n))$

31) Avalie assintoticamente as expressões abaixo e caracterize completamente cada caso usando o menor número de notações possível em cada item.

a) $O(n) + \Omega(n \log(n)) + \Theta(n)$

b) $O(n \log(n)) + \Theta(n) + (\Theta(n) * \Theta(n))$

c) $\Omega(n) + \Theta(n^2) + O(n^2) + (O(n)*O(n)*O(n))$

d) $\Omega(n) + O(n^2) + \Theta(n^2) + (\Theta(n)*\Theta(n)*\Theta(n))$

32) Use o Método mestre (quando possível) para resolver as seguintes equações de recorrência:

a) $T(n) = T(n/2) + \Theta(1)$

b) $T(n) = T(n/2) + \Theta(n)$

c) $T(n) = T(n/2) + \Theta(n) + \Theta(\log n) + \Theta(1)$

d) $T(n) = T(n/2) + 2^n$

e) $T(n) = 2T(n/2) + 2^n$

f) $T(n) = 2T(n/2) + \Theta(1)$

g) $T(n) = 2T(n/2) + \Theta(n)$

h) $T(n) = 2T(n/2) + \Theta(n^2)$

i) $T(n) = 4T(n/2) + \Theta(1)$

j) $T(n) = 4T(n/2) + \Theta(n)$

33) Montar a equação de recorrência de:

```
Algoritmo Pesquisa(vetor)
  if vetor.size() ≤ 1 then
    inspecione elemento;
  else
    inspecione cada elemento recebido (vetor);
    Pesquisa(vetor.subLista(1, (vetor.size() / 3) );
  end if
end.
```

34) Montar a equação de recorrência de:

Função fib (n)

se $n < 2$ **então**

retorne n

caso contrário

retorne $fib(n-1) + fib(n-2)$

35) Montar a equação de recorrência de:

```
Algoritmo mergesort_aux(ref, A, l, r)
    se (l < r) então
        m <- (l+r)/2;
        mergesort_aux(A, l, m)
        mergesort_aux(A, m + 1, r)
        intercala(A, l, m, r)
    fimse
fimalgoritmo
```

36) Resolva a seguinte relação de recorrência pelo método de substituição:

$$s(1) = 2$$
$$s(n) = 2s(n-1) \text{ para } n \geq 2$$

37) Considere as seguintes sub-rotinas:

a) As sub-rotinas sempre terminam? Explique.

b) Qual a complexidade dos algoritmos no pior caso?

c) Qual a complexidade dos algoritmos no melhor caso?

d) Quantos passos os algoritmos executam?

e) Qual problema os algoritmos se propõem a resolver?

f) As duas soluções são equivalentes ou possuem diferenças em sua semântica?

```c
int q(int v[], int N, int x) {
    int i;
    int p = -1;
    for (i=0; i < N; i++) {
        if (x == v[i])
            p = i;
    }
    return p;
};
```

```c
int b(int v[], int N, int x) {
    int i;
    for (i=0; i < N; i++) {
        if (x == v[i])
            return i;
    }
    return -1;
};
```

38) Considere as seguintes sub-rotinas:

```c
int a(int v[], int N, int x) {
    int i = N/2;
    int y = v[i];

    if (N == 0)
        return -1;
    else if (x == y)
        return i;
    else if (x < y)
        return a(v, i, x);
    else
        return a(&v[i+1], N-(i+1), x);
}
```

```c
int b(int v[], int N, int x) {
    int i;
    for (i=0; i < N; i++) {
        if (x == v[i])
            return i;
    }
    return -1;
};
```

a) Qual a complexidade de cada algoritmo?

b) As sub-rotinas sempre terminam? Explique.

c) Avaliar qual das duas soluções executam em um melhor número de passos.

d) As duas soluções resolvem o mesmo problema?

e) Comente sobre as disposições e/ou pressuposições do dado de entrada.

f) As duas soluções são equivalentes ou possuem diferenças em sua semântica?