



Aluno(a): _____

Primeiro trabalho (Parte 1)

- Coloque as seguintes funções em ordem crescente segundo a ordem de crescimento assintótica de complexidade (notação O). $f_1(n) = \frac{n^2}{100000}$, $f_2(n) = \sqrt{n} \lg n$, $f_3(n) = 99999n$, $f_4(n) = \lg(n!)$.
- Suponha a seguinte comparação de implementações de dois algoritmos em uma mesma máquina. O algoritmo A tem tempo de execução $64n^2$. O algoritmo B tem tempo de execução 2^n . Para quais valores de n o algoritmo B ganha do algoritmo A ?
- Mostre que:
 - $\frac{n}{2} \lg(\frac{n}{2}) = \Omega(n \lg n)$.
 - $n^2 + 1000n = O(n^2)$.
 - $2^{n+1} = \Theta(2^n)$.
- Sejam $f(n)$ e $g(n)$ funções positivas. Assinale V ou F e justifique brevemente.
 - $f(n) = O(g(n))$ implica $g(n) = O(f(n))$.
☐ V ☐ F _____
 - $f(n) + g(n) = \Theta(\min(f(n), g(n)))$.
☐ V ☐ F _____
 - $f(n) = O(g(n))$ implica $g(n) = \Omega(f(n))$.
☐ V ☐ F _____
 - $f(n) = \Theta(f(n/2))$.
☐ V ☐ F _____
- Prove a correção e analise a complexidade do algoritmo a seguir.

```
bubble(A[], n)
1 for i = 1 to n - 1
2   for j = 1 to n - i
3     if (A[j] > A[j+1])
4       Troque A[j] com A[j+1]
5 return A;
```

- Analise a complexidade dos seguintes algoritmos:

(a) $f(n)$

```
1 if n == 1
2   return 1
3 else
4   return f(n-1)+f(n-1)
```

(b) $\text{busca}(A[], \text{key}, \text{min}, \text{max})$

```
1 if max < min
2   return -1;
3 else
4   mid = min + ((max-min)/2)
5 if (A[mid] > key)
6   return busca(A, key, min, mid-1);
7 else if (A[mid] < key)
8   return busca(A, key, mid+1, max);
9 else
10  return mid;
```