1)

n³/1000 – 100n² – 100n + 3

n³, n², n, 1

Maior = N³

8)

O(1): Possui um tempo constante. O tempo é o mesmo para toda execução.

O(n): As operações aumentam de maneira linear de acordo com as entradas ao longo de uma única iteração para cada item.

O(n²): É conhecido como tempo quadrático, o que significa que toda vez que o número de elementos aumenta, aumentamos as operações quadraticamente.

O log(n): Um algoritmo logarítmico reduz pela metade uma lista toda vez que é executado.

O(2^n): O tempo exponencial é 2^n, em que 2 depende das permutações envolvidas. Esse algoritmo é o mais lento de todos.

9)

Levar em consideração apenas as repetições do código, Verificar a complexidade das funções ou métodos próprios da linguagem (se for utilizado), Ignorar as constantes e utilizar o termo de maior grau.

10)

Pa:

an = (a1 + (n - 1) \* r); O(n)  
Sn = (a1 + an)/2; O(n)

Pg:

an = (a1 \* pow(q,n-1)); O(n)  
Sn = ((a1\*(pow(q,n) - 1))/(q-1)); O(n)