Curso: Ciência da Computação
Unidade Curricular: Análise de Algoritmos
Ano/Período: 6º
Estudante: Marco Túlio Palhares Breyner

1- O(1)
O(log n)
O(n)
O(n log n)
O(n^2)
O(n^3)
O(2^n)
O(n!)

2- <u>O que significa dizer que uma função g(n) é O(f(n)) ?</u>
significa que f(n) é o limite superior de g(n)
<u>O que significa dizer que uma função g(n) é Ω(f(n)) ?</u>
significa que f(n) é o limite inferior de g(n)

4- <u>selection sort</u>:
O algoritmo percorre o vetor procurando o menor valor, quando acha ele troca com o menor valor anterior. O índice de comparação aumenta 1 e repete o processo.
custos:
pior caso: O(n^2)
caso médio: O(n^2)
melhor caso: O(n^2)

<u>insertion sort</u>:
O algoritmo assume o primeiro elemento como já ordenado e define seu sucessor como auxiliar. Usa o auxiliar comparando-o de trás para frente com os antecessores. Quando acha a posição correta ele insere o auxiliar e “empurra” o restante do vetor para frente.
custos:
pior caso: O(n^2)
caso médio: O(n^2)
melhor caso: O(n^2)

5- O problema abordado foi a "Multiplicação de Inteiros". A forma tradicional x*y tem complexidade $O(n^2)$. Para otimizar com Divider and Conquer, usamos o Algoritmo de Karatsuba. Etapas técnicas:
Dividir: Separamos os dois números grandes em parte alta e parte baixa. Se o número possui n dígitos, o corte é feito na posição m = n/2

Conquistar: Usamos o algoritmo de Karatsuba, que ao invés de 4 multiplicações recursivas, faz 3 chamadas recursivas

1. O produto das partes altas (Z0);
2. O produto das partes baixas (Z2);
3. O produto da soma das partes (Z1).

Combinar: Alcançamos o resultado final aplicando os resultados parciais na fórmula: $X \times Y = Z_2 \times 10^{(2m)} + Z_1 \times 10^m + Z_0$