Nome: Guilherme Kwaczynski Trajanoski

Exercícios de Complexidade

1- a) A eficiência de tempo de um algoritmo se refere à quantidade de tempo que o algoritmo leva para executar em relação ao tamanho da entrada.

b)O número de operações executadas, o uso de memória e detalhes específicos do hardware.

c) As grandezas utilizadas são o número de operações executadas e o uso de memória por parte do algoritmo.

d) Se abstrairmos dos detalhes específicos do hardware, como a velocidade exata da CPU ou a arquitetura específica do sistema.

e) É utilizada a notação Big O para expressar a complexidade de tempo de um algoritmo, pois verifica sem tanta rigidez o máximo de recursos que o algoritmo vai utilizar.

2- Uma das possíveis formas de se descrever a complexidade de um algoritmo é a chamada Notação-Big-Oh. Explique o que você entendeu por esta definição.
A notação Big-O descreve a ordem de grandeza da complexidade de um algoritmo, ignorando fatores constantes e termos de menor ordem. Ela é usada para dar uma visão geral do desempenho relativo de algoritmos para conjuntos de dados grandes. A notação Big-O é representada por O(f(n)), onde "f(n)" é uma função que descreve a taxa de crescimento do algoritmo em relação ao tamanho da entrada "n".

3- Neste algoritmo, há dois loops aninhados, com o primeiro loop, com índice i, executa n vezes e o segundo loop, com índice j, depende do valor de i. Se for ímpar, então o segundo loop é executado 2n vezes, senão, o segundo loop executa apenas n vezes.
Portanto a complexidade de O(n2) representa o pior caso.

4- O pior caso ocorre quando o elemento procurado não está presente no vetor ou está na última posição. Neste caso, a busca linear percorre todo o vetor até o final. A complexidade de tempo no pior caso é, portanto, O(n), onde n é o número de elementos no vetor.
O melhor caso ocorre quando o elemento procurado está na primeira posição do vetor. Neste caso, a busca linear encontra o elemento imediatamente na primeira iteração. A complexidade de tempo no melhor caso é O(1).