

## Revisão para Prova Teórica

1. Defina os seguintes conceitos relacionados a Grafos:

- Vértice (ou Nó):
- Aresta:
- Grafo Direcionado:
- Grafo Não Direcionado:
- Grafo Ponderado:

2. Explique a diferença entre a representação de grafos por Matriz de Adjacência e por Lista de Adjacência, citando vantagens e desvantagens de cada uma.

3. Descreva o funcionamento do algoritmo de Busca em Largura (BFS) em grafos. Em que tipo de problema ele é mais adequado?

4. Explique o objetivo e o funcionamento do Algoritmo de Dijkstra. Qual é a sua principal aplicação?

5. Descreva o funcionamento do algoritmo de Busca em Profundidade (DFS) em grafos. Em que tipo de problema ele é mais adequado?

6. Qual a principal diferença entre Pesquisa Sequencial e Pesquisa Binária? Em que condições a Pesquisa Binária é mais eficiente?

7. Descreva o funcionamento do algoritmo de ordenação Bubble Sort. Qual a sua complexidade de tempo no pior caso?

8. Explique o princípio de "dividir para conquistar" aplicado ao algoritmo Merge Sort. Qual a sua complexidade de tempo?

9. Defina o conceito de Tabela Hash (ou Tabela de Dispersão) e explique sua principal finalidade na organização de dados.

10. Explique o que é a complexidade de tempo e a complexidade de espaço de um algoritmo. Por que é importante analisar a eficiência de algoritmos?

Questões extra:

11. Dado que o algoritmo Quick Sort tem complexidade média de  $O(n \log n)$  e o algoritmo Bubble Sort tem complexidade  $O(n^2)$ , qual afirmação é correta?

- A. O Quick Sort é sempre mais rápido, independentemente do tamanho do array.
- B. O Bubble Sort pode ser mais eficiente para listas totalmente desordenadas.

- C. O Quick Sort tende a ser mais eficiente para grandes conjuntos de dados.
- D. Ambos têm a mesma eficiência quando o array já está ordenado.
- E. O Bubble Sort consome menos memória e, por isso, é mais eficiente.

3. Qual dos seguintes algoritmos é considerado não estável?

- A. Insertion Sort
- B. Merge Sort
- C. Counting Sort
- D. Selection Sort
- E. Bubble Sort

4. A complexidade de espaço do Merge Sort é  $O(n)$  porque:

- A. Ele utiliza árvores binárias auxiliares para ordenação.
- B. Ele precisa armazenar todos os elementos na memória secundária.
- C. Ele cria subarrays temporários em cada etapa de intercalação.
- D. Ele realiza muitas trocas que consomem espaço adicional.
- E. Ele aloca espaço fixo, independente da entrada.

5. Após a primeira iteração completa do algoritmo Insertion Sort, considerando a entrada [7, 3, 5, 1, 2], qual será a nova sequência?

- A. [3, 7, 5, 1, 2]
- B. [3, 5, 7, 1, 2]
- C. [7, 3, 5, 1, 2]
- D. [3, 5, 1, 2, 7]
- E. [7, 5, 3, 1, 2]

6. Qual algoritmo apresenta desempenho linear ( $O(n)$ ) no melhor caso, quando o array já está ordenado?

- A. Bubble Sort com otimização
- B. Merge Sort
- C. Counting Sort
- D. Heap Sort
- E. Selection Sort

7. Cite uma vantagem do Insertion Sort sobre o Selection Sort.