Módulo 07 Estrutura de Dados
Aula 07: Grafos

Questões de Aprendizagem

1) Implementação de Algoritmo BFS

Implemente o algoritmo de Busca em Largura (BFS) para um grafo representado por uma lista de adjacência.

O algoritmo deve ser capaz de:
- Partir de um vértice inicial dado.
- Visitar todos os vértices alcançáveis a partir do vértice inicial, seguindo a ordem de menor número de arestas possível (largura antes de profundidade).
- Imprimir a ordem de visitação dos vértices.

Entrada: Uma lista de adjacências representando o grafo. Um vértice inicial.

```
grafo = {0: [1, 2], 1: [2], 2: [0, 3], 3: [3]}
vértice_inicial = 2
```

Saída: A ordem de visitação dos vértices a partir do vértice inicial.

```
2 0 3 1
```

Dica: Utilize uma fila para controlar os vértices a serem visitados e um conjunto para marcar os vértices já visitados.

Módulo 07 Estrutura de Dados
Aula 07: Grafos

Questões de Aprendizagem

2) Detecção de Ciclo em Grafo Dirigido

Escreva uma função em Python que detecte a presença de um ciclo em um grafo dirigido. O grafo será representado por uma lista de adjacências, e sua função deve:

Determinar se o grafo contém pelo menos um ciclo. Retornar True se um ciclo for encontrado e False caso contrário.

Entrada: Uma lista de adjacências representando o grafo dirigido.

```
grafo = {0: [1], 1: [2], 2: [0, 3], 3: []}
```

Saída: Um valor booleano indicando a presença de um ciclo.

```
True
```

Dica: Considere usar uma abordagem de busca em profundidade (DFS) com um conjunto para marcar os vértices visitados e outro conjunto para acompanhar a pilha de recursão. Um ciclo é encontrado se um vértice já está na pilha de recursão quando tentamos visitá-lo novamente.