

Exercícios

Use a classe Grafo disponível no link a seguir como modelo.

https://github.com/professorbossini/20251_maua_cic401

Implemente uma classe Java chamada **Grafo** que realiza uma **busca em profundidade (DFS)** a partir do vértice 0. O grafo é **não direcionado** e deve ser representado por **lista de adjacências** usando um vetor de `LinkedList<Integer>`. A entrada será fornecida **via linha de comando (args)**, da seguinte forma:

- O primeiro valor representa o número de vértices do grafo.
- Os valores seguintes devem ser interpretados como pares de vértices conectados por uma aresta.

Exemplo de execução:

```
//java Grafo 6 0 1 0 2 1 3 2 4 4 5
```

Explicação da entrada:

- 6 → número de vértices (vértices de 0 a 5)
- 0 1, 0 2, 1 3, 2 4, 4 5 → arestas entre os vértices

O programa deve realizar a DFS a partir do vértice 0 e imprimir a ordem dos vértices visitados.

Saída esperada:

```
DFS a partir do vértice 0:  
0 1 3 2 4 5
```

Dica: utilize um vetor `boolean[] visitado` para controlar quais vértices já foram visitados durante a execução do algoritmo.

Implemente uma classe Java chamada **Grafo** que realiza uma **busca em largura (BFS)** para encontrar a **distância mínima** (menor número de arestas) entre dois vértices. A entrada será fornecida **via linha de comando (args)**, da seguinte forma:

- O primeiro valor representa o número de vértices do grafo.
- Os valores seguintes devem ser interpretados como pares de vértices conectados por uma aresta.
- Os dois últimos valores indicam o vértice de origem e o vértice de destino.

Exemplo de execução:

```
//java Grafo 6 0 1 0 3 1 5 2 5 3 4 4 5 0 5
```

Explicação da entrada:

- 6 → número de vértices (de 0 a 5)
- 0 1, 0 3, 1 5, 2 5, 3 4, 4 5 → arestas
- 0 5 → origem = 0, destino = 5

Nesse grafo, existem dois caminhos possíveis de 0 até 5:

- 0 → 1 → 5 (2 arestas)
- 0 → 3 → 4 → 5 (3 arestas)

O algoritmo deve encontrar o caminho mais curto (com menos arestas).

Saída esperada:

Distância mínima de 0 até 5 é 2

Referências

BONDY, J. A.; MURTY, U. S. R. **Graph theory**. New York: Springer, 2008. (Graduate Texts in Mathematics, v. 244).

CORMEN, Thomas H. et al. **Introduction to Algorithms**. 3. ed. Cambridge: MIT Press, 2009.

FEOFILOFF, Paulo. **Análise de Algoritmos**. Disponível em: https://www.ime.usp.br/~pf/analise_de_algoritmos/. Acesso em: março de 2025.

KLEINBERG, Jon; TARDOS, Éva. **Algorithm Design**. Boston: Pearson, 2006.