UFMA - Universidade Federal do Maranhão
São Luís, 23 de junho de 2024
Dupla: Maxwell Pires Silva e Grace Kelly Barros Araujo dos Santos

Estrutura de Dados II

Relatório - Trabalho de Implementação II

Introdução

O presente relatório visa demonstrar a lógica e os códigos utilizados para resolver as questões propostas no trabalho de implementação dos Algoritmos de busca e visitação em Grafos.

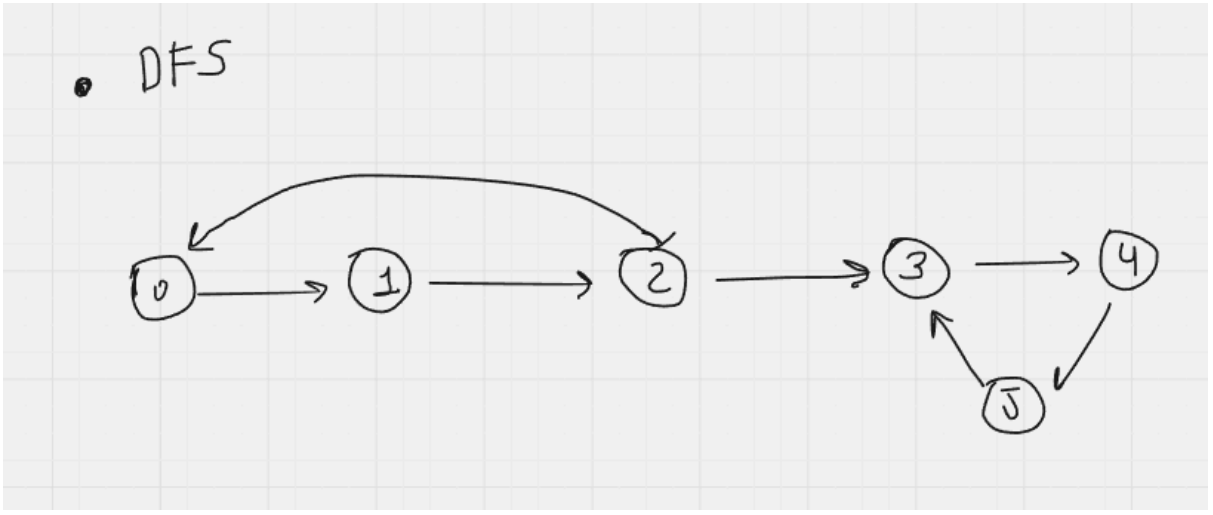
Objetivos

Implementar algoritmos de Grafos em Java, produzir um vídeo da resolução e explicação:
Os vídeos foram hospedados no YouTube como Não Listado:
Link problema 1 e 2: https://youtu.be/tu4__-XE6VM
Link problema 3 e 4: https://youtu.be/aNdiTbTvI-Y

Problema 1

Foi solicitado a implementação do Algoritmo de DFS (Busca em Profundidade). No código foi gerado uma classe Grafo, com as funções de adicionar aresta, contar aresta de um vértice até outro, verificar caminho e imprimi-lo, e por fim imprimir todos as arestas de retorno.

O Grafo de exemplo utilizado foi esse, bem simples mas de fácil verificação para o funcionamento do Algoritmo.

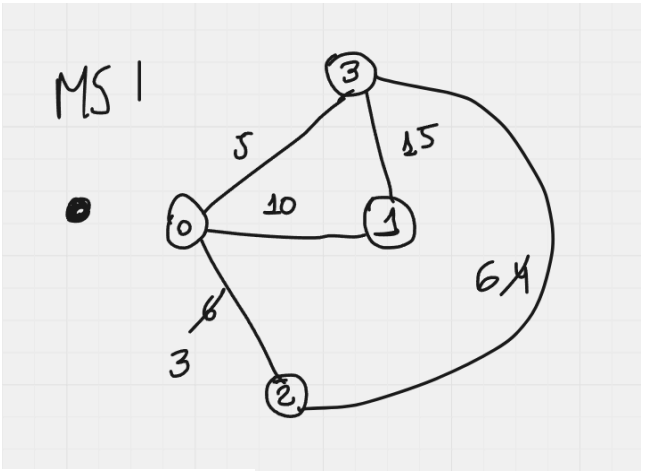



```
Número de arestas de 0 a 4: 4
Caminho de 0 a 3: [0, 1, 2, 3, 4]
Arestas de retorno:
Aresta de retorno: 2 -> 0
Aresta de retorno: 5 -> 3
```

Foi solicitado uma DFS entre o 0 e 4. Podemos notar que há 4 arestas, e logo abaixo as arestas impressas como resultado do caminho. Por fim, as arestas de retorno são calculadas de forma independente aos vértices de origem e destino, pois encontras as arestas de retorno para todo o grafo.

Problema 2

Foi solicitado criar uma versão modificada do MST, mas agora com ordenação decrescente das arestas, testando se a retirada de cada aresta desconecta ou não o grafo, e provar que ao final realmente temos uma MST. A ordem de crescimento do número de comparações de peso de aresta na implementação do algoritmo produzido é O(E^2)no pior caso, enquanto que para Kruskal é O(ElogE), tornando Kruskal geralmente eficiente. O algoritmo foi testado no grafo abaixo.

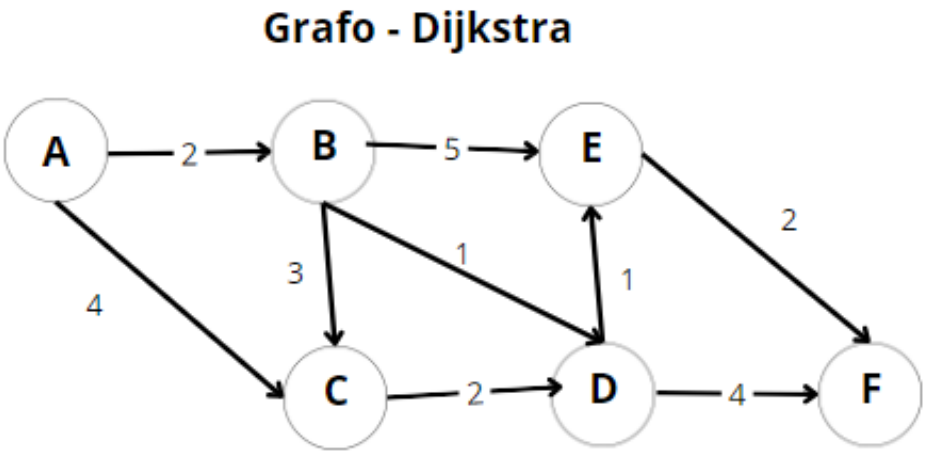



```
Arestas na MST:
0 -- 1 == 10
0 -- 3 == 5
0 -- 2 == 3
```

E as áreas encontradas foram essas, que realmente correspondem à MST, com a ressalva de que o código precisa percorrer E^2 vezes no pior caso para encontrar a MST.

Problema 3

O problema 3 se trata de uma implementação modificada do algoritmo de Dijkstra, para retornar os dois menores caminhos entre um par de vértice (u,v). O grafo utilizado na explicação e execução do código foi:

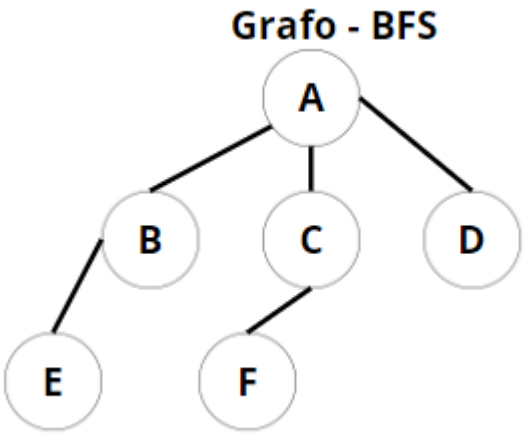


Resultado: Saída com o menor caminho e a alteração realizada, que também guarda e imprime o segundo menor caminho.

```
Selecione uma opção: 3
Para retornar os dois menores caminhos entre um par de vértices
Insira o vértice de Origem:
A
Insira o vértice de Destino:
D
Menor caminho: A -> B -> D : 3
Segundo menor caminho: A -> C -> D : 6
```

Problema 4

Foi realizada a implementação do algoritmo BFS, utilizando o seguinte grafo como exemplo para execução:



Como pedido, foi feita a adição dos seguintes métodos:

a - Retornar o número de arestas entre um par de vértices.

```
-----  BUSCA EM LARGURA -----

[1] - Retornar número de arestas entre Origem e Destino:
[2] - Retornar o caminho de Origem ao Destino:
[3] - Retornar vérticecs a uma distância D:
[4] - Voltar.
1
Insira o vértice de Origem:
A
Insira o vértice de Destino:
F
O número de arestas existentes entre A e F é: 2
```

b - Retornar o caminho de origem e destino.

```
2
Insira o vértice de Origem:
E
Insira o vértice de Destino:
F
Caminho de E a F
E
B
A
C
F
```

c - Retornar vértices a uma distância D.

```
3
Insira o vértice de Origem:
F
Insira a distância:
4
E
```

Conclusão

Implementamos vários algoritmos fundamentais na teoria de grafos e realizamos alterações que foram importantes para um entendimento mais aprofundado sobre eles. A busca em profundidade e a busca em largura foram implementadas com métodos adicionais para retornar o número de arestas entre vértices, o caminho de origem a destino, identificar arestas de retorno (DFS) e listar vértices que estão a uma dada distância (BFS). Foi desenvolvida uma nova versão do algoritmo da Árvore Geradora Mínima (MST), além disso, foi realizada a modificação do algoritmo de Dijkstra para retornar os dois menores caminhos entre pares de vértices.