

PMR2300 – Exercícios sobre grafos

Prof. Fabio Cozman

1) Considere um grafo direcionado ponderado (ou seja, cada aresta tem um peso) com 5 nós, e armazenado em uma matriz de adjacências como segue. O valor v_{ij} na linha i e coluna j da matriz indica que a aresta do i ésimo nó para o j ésimo nó tem peso v_{ij} no caso de v_{ij} maior que zero; caso contrário não existe aresta. A matriz de adjacências é:

0	5	6	0	0
5	0	4	2	0
0	0	0	2	5
0	3	2	0	1
0	0	0	0	0

Desenhe o grafo indicando os nós e arestas, e obtenha um menor caminho entre o primeiro e o quinto nós usando o algoritmo de Dijkstra (indique o valor da distância D para todos os nós após o processamento de cada nó).

2) Dado um grafo não-direcionado e um nó A do grafo, apresente um algoritmo que receba o grafo e o nó A e retorne uma árvore que contenha todos os nós do grafo e que tenha A como raiz (assuma que existe um caminho entre A e qualquer outro nó; ou seja, o grafo é inteiramente conectado). O algoritmo deve ser comentado e suficientemente claro para que seja possível codificá-lo sem dúvidas nem ambiguidades). [1.5] Apresente a complexidade do algoritmo desenvolvido em notação BigOh no pior caso.

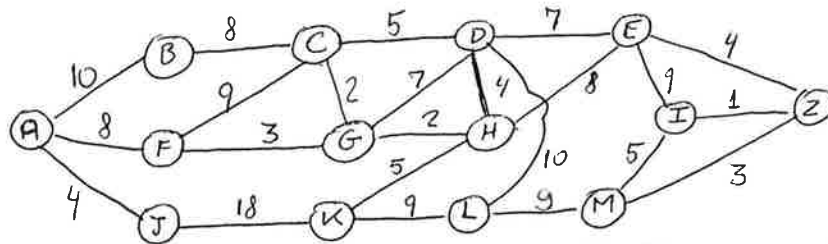
3) Considere o código abaixo. Determine a complexidade em notação BigOh, no pior caso, justificando cada passo. Adote:

- Cada objeto da classe Grafo contém um arranjo de objetos do tipo Nó; o número de nós pode ser obtido com a função númeroNós.
- Cada objeto da classe Nó contém um nome (String) e um identificador (int).
- A função algoritmoDijkstra recebe um grafo e um nó, e retorna um arranjo (com tamanho igual ao número de nós do grafo) com os valores da função D do algoritmo de Dijkstra a partir do nó.
- A função ordenaPorInserção recebe um arranjo e o ordena por inserção.
- O número de nós do grafo g é igual ao tamanho do arranjo a ; use N para indicar esse número.

É possível modificar o código de maneira que a saída seja igual mas a complexidade seja menor (em notação BigOh no pior caso)? Caso afirmativo, apresente código; caso negativo, justifique.

```
int sub(Grafo g, int a[], int b) {
    int sum = 0;
    int c[];
    for (int i=0; i<g.númeroNós(); i++) {
        if (g.nó(i).nome.compareTo("X") {
            c = algoritmoDijkstra(g, g.nó(i));
            ordenaPorInserção(c);
        } else {
            a[i] = a[i] + g.nó(i).identificador;
        }
    }
    if (b<0)
        return (-1);
    for (int j=b; j<a.length * c.length; j++)
        for (int k=j; k<a.length; k++)
            sum = sum + a[j] * c[k];
    return(sum);
}
```

4) No grafo abaixo, cada nó representa uma máquina, e cada aresta representa o tempo gasto para uma peça ser levada de uma máquina a outra e executar a operação na máquina de chegada. Usando o algoritmo de Dijkstra, obtenha o menor tempo possível para uma peça sair da máquina A e ser processada na máquina Z (indique os resultados intermediários do algoritmo após o processamento de cada nó).



5) A seguinte classe foi criada para representar um grafo direcionado:

```
public class GrafoDirecionado {
    String nome;
    NÓGrafo nós[]; }

```

onde cada nó é identificado por seu índice no arranjo nós, e cada nó pertence à seguinte classe:

```
public class NÓGrafo {
    int dado;
    NÓListaLigada filho; }

```

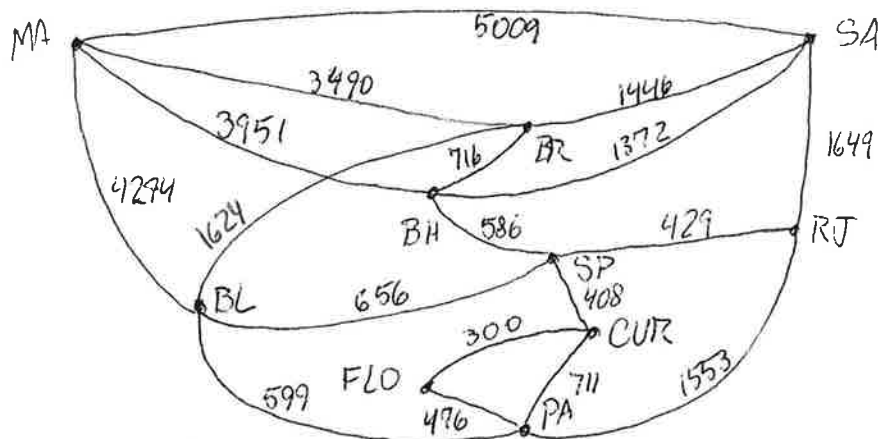
onde a classe NÓListaLigada implementa um nó de lista ligada conforme visto em aula. A ideia é que cada nó do grafo armazena um dado e uma lista ligada contendo os nós filhos (ou seja, cada nó da lista ligada contém um filho). Escreva uma função em Java (comentando o código em detalhes) com chamada

```
boolean[][] converteGrafo(GrafoDirecionado g)

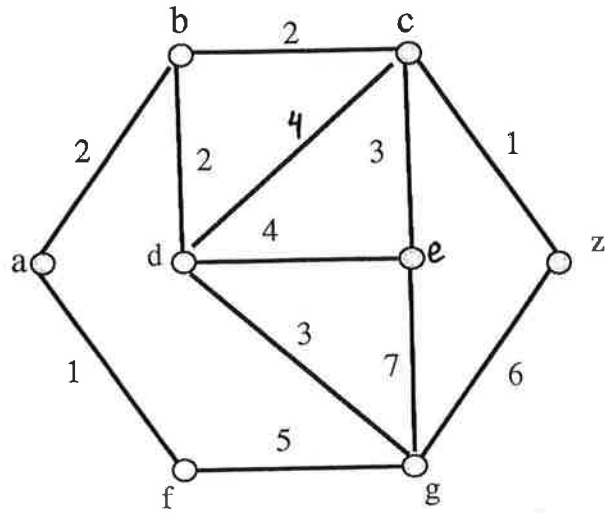
```

que recebe um objeto Grafo e retorna uma matriz de adjacências na qual a célula (i,j) contém true se existe uma aresta do iésimo para o jésimo nó, e false caso contrário. Apresente o custo computacional do seu código em notação BigOh no pior caso; justifique.

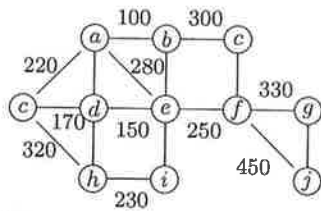
6) Considere o grafo abaixo, e obtenha o caminho de menor distância entre SP e cada uma das outras cidades, justificando cada passagem do algoritmo utilizado.



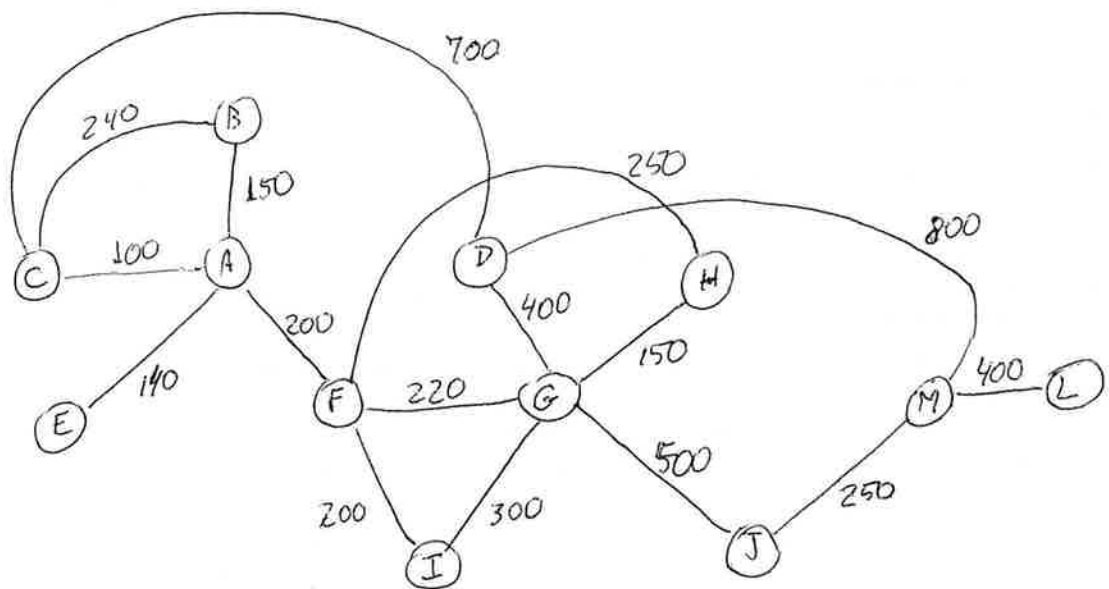
7) Considere o grafo não-dirigido ponderado abaixo. Qual é o caminho de menor custo entre a e z? Ao aplicar o algoritmo de Dijkstra com destino z, a função auxiliar D contém o valor estimado de custo até o passo corrente, e o conjunto C guarda os nós já processados. Qual é o valor de D(a) e D(f) no momento imediatamente anterior à colocação do nó e em C?



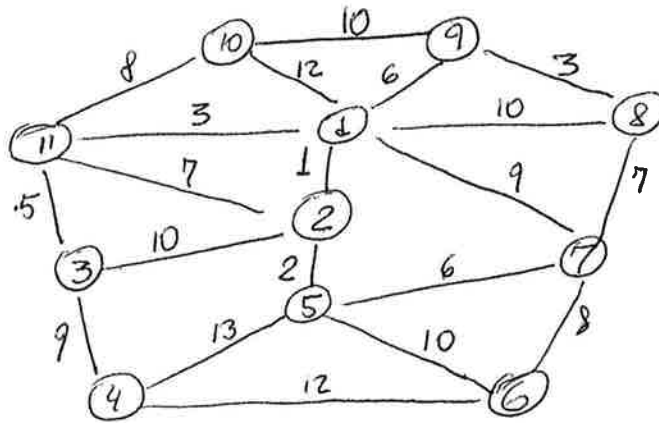
8) Considere o grafo não-direcionado abaixo, onde todas as arestas verticais tem custo 200. Qual é o caminho de menor custo entre b e j?



9) Considere o grafo não-direcionado abaixo. Qual é o caminho de menor custo entre os nós A e L?



10) Considere o grafo abaixo. Obtenha uma árvore contendo todos os nós, e todas as arestas que indicam caminho mínimo do nó 1 para qualquer outro nó; mostre o valor da função $D(\cdot)$ associada a cada nó ao fim do algoritmo de Dijkstra. [1.5] A árvore obtida é uma árvore geradora mínima? Justifique. [1.0]



11) Considere o grafo ponderado abaixo. Apresente uma árvore que indica todos os caminhos mínimos entre o nó A e todos os demais nós. [1.0] Apresente uma árvore geradora mínima. [0.5]

