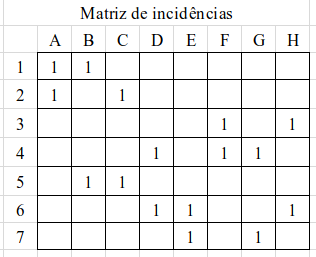
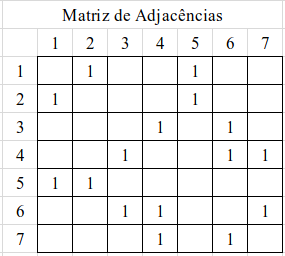
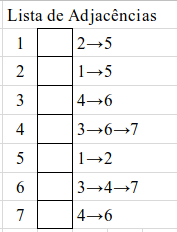
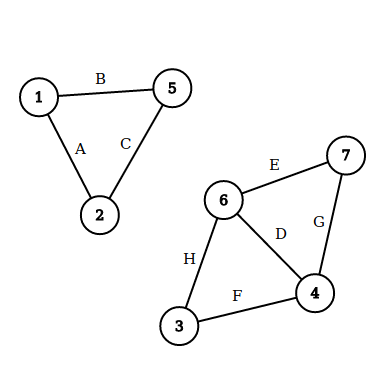
**Lista 01 - GRAFOS 03/10/2018**

01) Desenho do grafo G:



ordem(G) = 7

tamanho(G) = 8  
Δ(G) = 3

δ(G) = 2  
w(G) = 2

G não é uma floresta, pois uma floresta não possui ciclos.

G não é conexo, pois não existe um caminho entre quaisquer par de seu vértices. Não existe, por exemplo, um caminho entre os vértices 1 e 7.A cintura do grafo é: g(G) = 3 (3-4-6-7, por exemplo)

Trilha de comprimento 5: **6-4-7-6-3-4**

Circuito de comprimento 5: **não existe circuito de comprimento 5!**

02) Escreva um algoritmo que receba a lista de adjacências de um grafo e dois vértices u e v e então remova a aresta uv da lista de adjacências (se ela existir no grafo, obviamente).

Obs: A lista ligada de adjacências é um vetor de listas ligadas e a lista ligada é uma estrutura que contém o valor e o próximo.

Algoritmo removendo\_arestas

Entrada: Uma lista de adjacência V de um grafo, os vértices u e v

Saída: A lista de adjacência V sem a aresta uv

se u != v

V = remove\_aresta(V, u, v)

V = remove\_aresta(V, v, u)

devolva V e pare

remove\_aresta(V, v1, v2)

nó = V[v1]

enquanto (nó != nulo e nó.valor < v2)

se nó.proximo.valor != v2

nó = nó.proximo

se (nó != nulo e nó.proximo = v2)

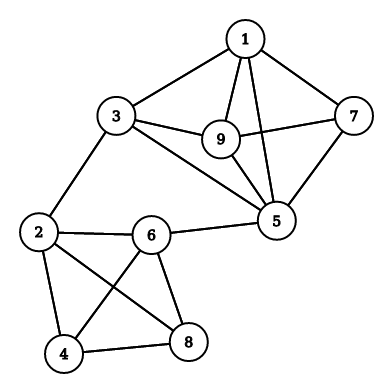
r = nó.proximo

nó.proximo = r.proximo

libere a memoria de r

devolva V

03) O grafo G representado pela lista de adjacências possui a seguinte estrutura:



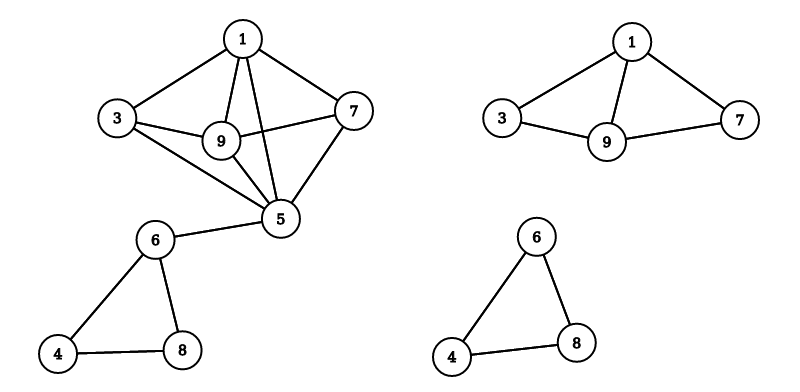
Aplicando o algoritmo k-vértice-conexo obtemos:

n = 9

i = 1, 2, 3

k = 3

Removendo o vértice 2, percebemos que os vértices 5 e 6 tornam-se vértices de corte. Dessa forma, removendo o vértice 5, por exemplo, desconectamos o grafo G gerando o grafo G’ com duas componentes conexas.



Logo, concluímos que o grafo G é 2-vértice-conexo

Aplicando o algortimo k-aresta-conexo obtemos:

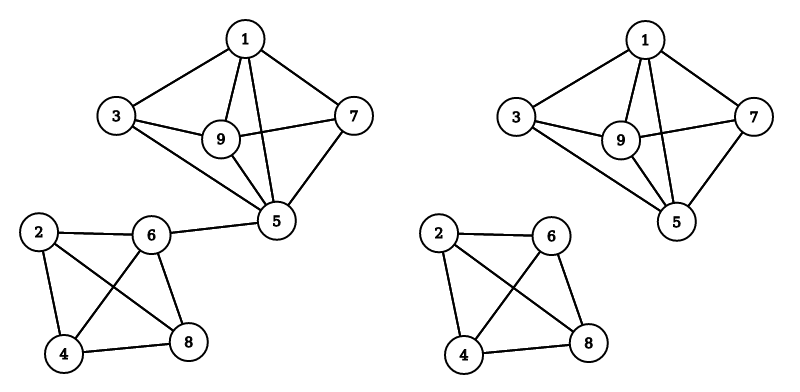
m = 17

i = 1, 2, 3

k = 3

Removendo a aresta que liga os vértices 2 e 3, obtemos uma ponte ligando os vértices 5 e

6. Dessa forma, removendo a ponte desconectamos o grafo, gerando o grafo G’ com duas componentes conexas.



Logo, concluímos que o grafo G é 2-aresta-conexo, pois precisamos remover no mínimo duas arestas para desconectá-lo.

**Portanto, x=2 e y=2.**

04) Escreva um algoritmo que receba um grafo e devolva a quantidade de componentes conexas do grafo (*sugestão*: adapte o algoritmo *floresta*).

Entrada: Um grafo G com n vértices

Saída: Quantidade de componentes conexas de G

se n = 0

devolva 0 e pare

componentes = 0

para i = 1 até n

visitado[i] = falso

crie uma fila F com n posições

enquanto houver vértice v não visitado (visitado[v] = falso)

componentes = componentes + 1

insira v em F

visitado [v] = verdadeiro

enquanto F não for vazia

remova de F obtendo u

para cada vértice w adjacente a u

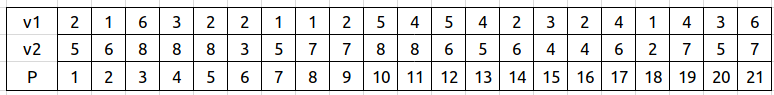
se visitado [w] = falso

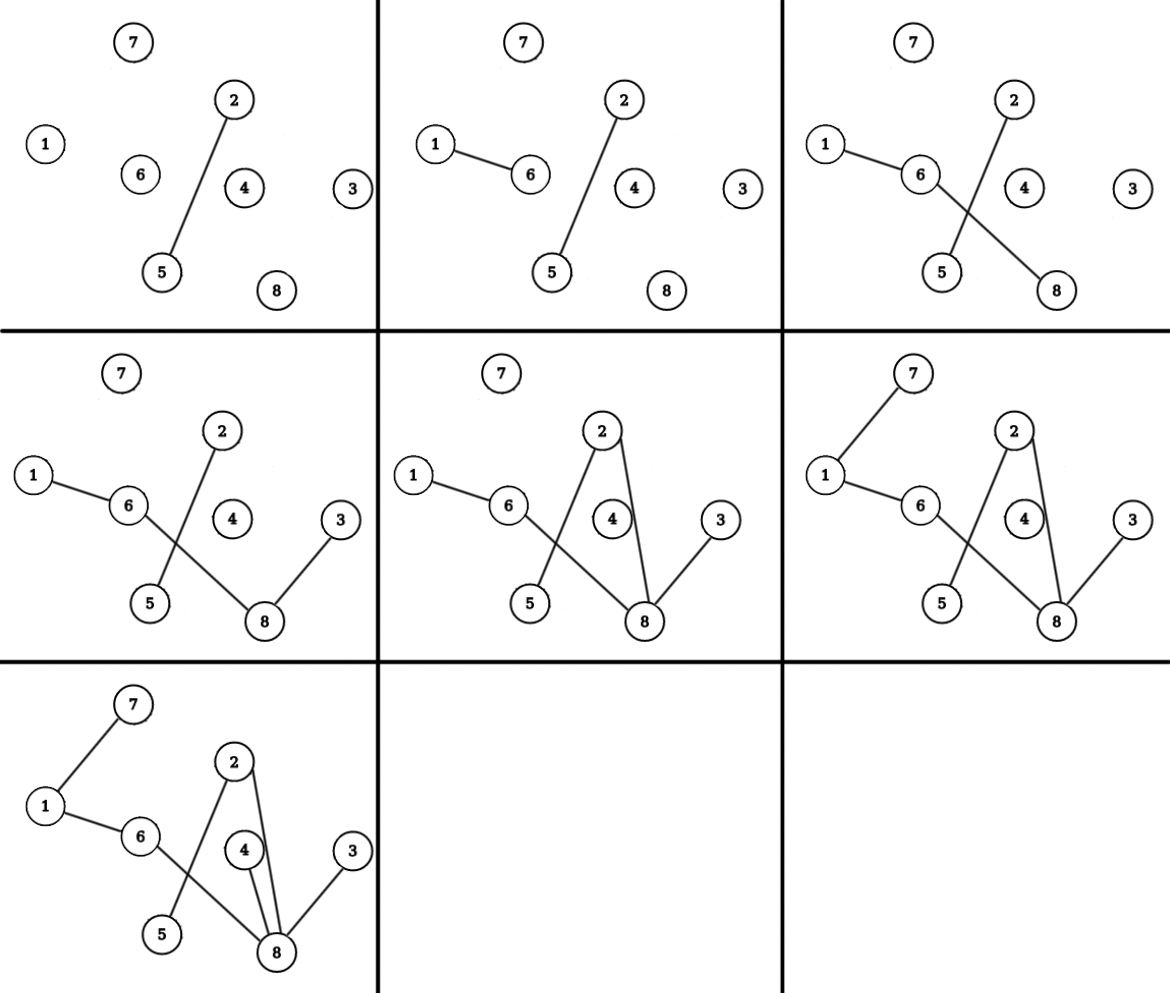
insira w em F

visitado [w] = verdadeiro

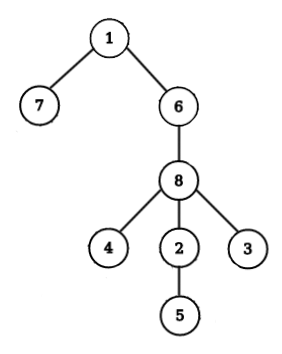
devolva componentes

05) Obtemos a seguinte tabela com as arestas ordenadas por ordem de custo:

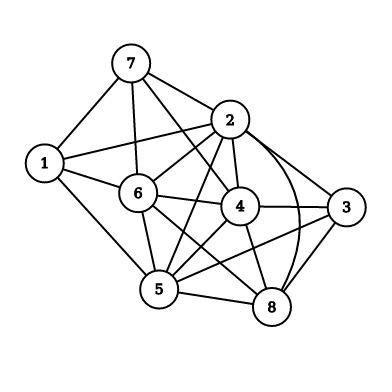


Posteriormente, aplicando o algoritmo de Kruskal, distribuímos as arestas no grafo, obtendo a sequência: 

Dessa forma, obtemos árvore geradora de custo mínimo abaixo, que possui custo = 34



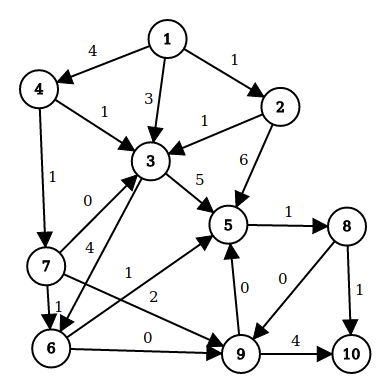
06) O grafo representado na questão 5 possui a seguinte estrutura (os pesos não foram representados para não sobrecarregar o desenho):



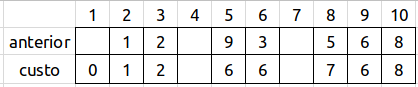
Aplicando o algoritmo de Fleury, podemos encontrar a seguinte Trilha de Euler:

**2-5-8-3-4-7-1-6-8-4-2-7-6-2-1-5-6-4-5-3-2-8**

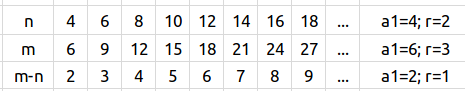
07) O grafo representado pela matriz de incidências possui a seguinte estrutura:



Aplicando o algoritmo de Dijkstra, obtemos:

  
Logo, o caminho mínimo entre v1 e v10 é: **1-2-3-6-9-5-8-10** e o custo desse caminho é 8

8)

  
Um grafo G é 3-regular desde que atenda as progressões aritméticas acima. Podemos provar a relação entre quantidade de vértices e arestas por indução.   
4+6+8+...+3n/2 = m; seja n par e ∀n≥4

Corolário: A quantidade de vértices de grau ímpar num grafo é sempre par. Portanto, todo grafo 3 regular sempre possui quantidade de vértices par.

Teorema 1: Um grafo é 2-aresta-conexo se e somente se não possui ponte.

Teorema 2: A aresta-conexidade de um grafo G é no máximo δ(G).

Um grafo 3 regular não possui ponte, pois todos os seus vértices estão ligados por 3 arestas, logo ele é 2-aresta-conexo.

Um grafo 3 regular também é 3-aresta-conexo, pois removendo 2 de suas arestas não é possível desconectá-lo. Para desconectar um grafo 3-regular, é necessário remover no mínimo 3 arestas, logo um grafo 3 regular não é 4-aresta-conexo. Percebemos que todo grafo 3-regular é no máximo 3-aresta-conexo, pois seu grau mínimo é 3.