

## **Trabalho 2 – Árvore Geradora Mínima**

**Prazo de entrega: 30/04/2014 no SSP**

Dado um grafo conexo ponderado  $G$ , imprima sua árvore geradora mínima em **ordem decrescente** de pesos. Utilize o algoritmo de Prim ou Kruskal.

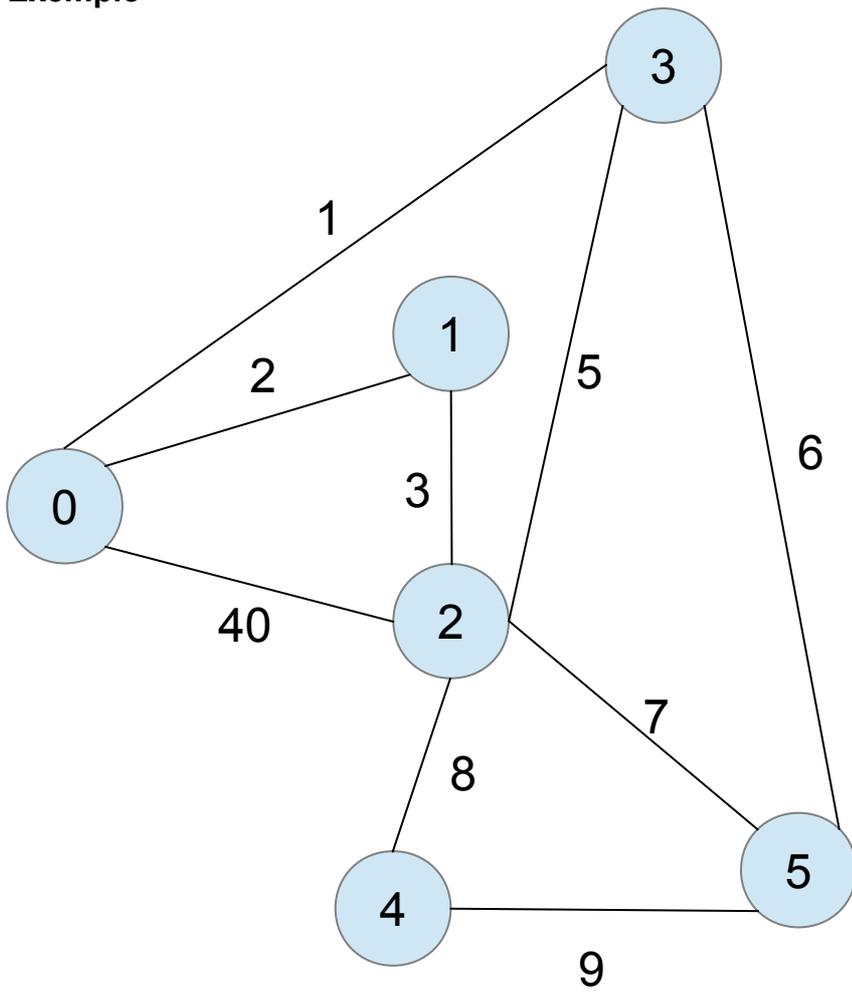
### **Entrada**

Os dados – fornecidos na entrada-padrão – contém apenas um caso de teste, que consiste nas informações do grafo de entrada.

A primeira linha de entrada contém dois números inteiros, **V** e **A**, que indicam respectivamente o número de Vértices e o número de Arestas do grafo.

As **A** linhas seguintes contém dois números inteiros, **V1** e **V2**, vértices que devem ser conectados no grafo (de **V1** para **V2**).

**Exemplo**



6	9	
0	3	1
0	1	2
2	1	3
0	2	4
2	3	5
3	5	6
2	7	5
2	4	8
4	5	9

## Saída

A saída - apresentada do dispositivo padrão consiste em várias linhas no formato:

**(<v1>,<v2>):<peso>**

Imprima as arestas em **ordem decrescente de peso**.

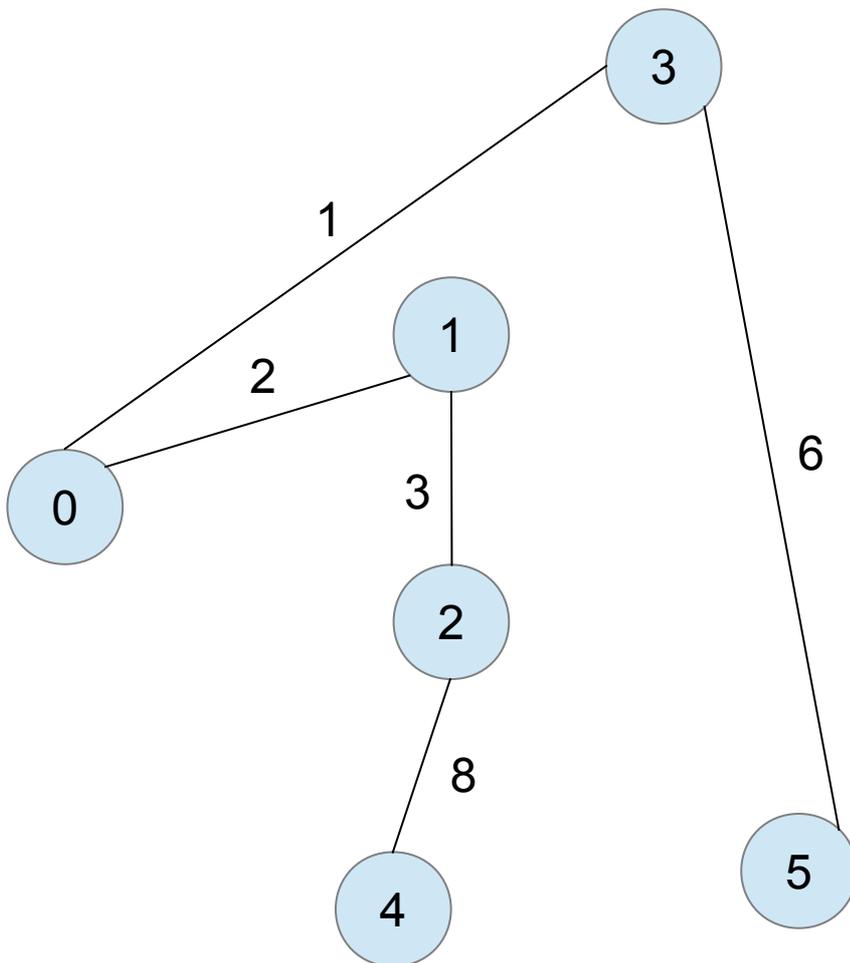
Para padronizar a saída no SSP, imprima primeiro o vértice cujo índice é menor.

Possível solução:

```
if (v1 > v2)
```

```
    troca(&v1,&v2); //troca o valor das variáveis
```

No caso do exemplo, a saída será:



**Terminal:**

(2,4):8

(3,5):6

(1,2):3

(0,1):2

(0,3):1

## Outras Informações Importantes

- O trabalho deve ser feito em duplas.
- O programa pode ser feito na linguagem C ou C++.
- Todas as submissões são checadas para evitar cópia/plágio/etc. Então, evite problemas e implemente o seu próprio código.
- Comente o seu código com uma explicação rápida do que cada função, método ou trecho importante de código faz (ou deveria fazer). Os comentários serão checados e valem nota.
- Mantenha a modularização entre código e dados, ou seja, divida o código explicitamente entre estruturas de dados (grafo) e algoritmos de manipulação. A modularização será checada e vale nota.
- Entradas/saídas devem ser lidas/escritas a partir dos dispositivos padrão, ou seja, use as funções “*printf(...)*” e “*scanf(...)*” / “*cin>>*” “*cout<<*”. Para testar, arquivos podem ser redirecionados para/de seu programa na linha de comando utilizando os operadores < e >.
- Exemplo:

```
# ./trab0 < entrada.txt > saida.txt
```