

Estruturas de Dados

PA04 - Conjuntos Disjuntos

Professores: Wagner Meira Jr
Eder Figueiredo

Roteiro da atividade

Nesta atividade você deverá implementar um programa que:

1. Lê da entrada padrão:
 - ☐ Inteiros n e m indicando respectivamente a quantidade de vértices e arestas.
 - ☐ Seguido de m linhas contendo os dados das arestas. Cada uma dessas linhas possui três inteiros u , v , c , indicando os vértices u e v que compõem a aresta de custo c .
2. Calcule utilizando o algoritmo visto na aula 18 o valor da árvore geradora mínima e o imprima na tela.

Exemplo

Entrada:

4 5

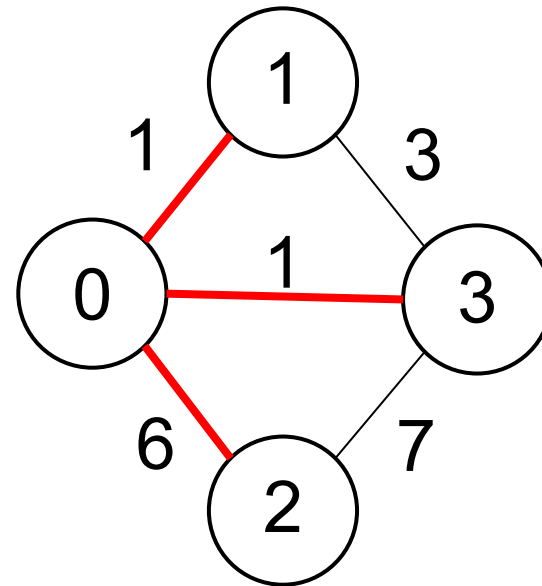
0 1 1

0 2 6

0 3 1

1 3 3

2 3 7



Saída esperada:

8

Exemplo

Entrada:

6 6

0 1 10

1 2 1

1 5 1

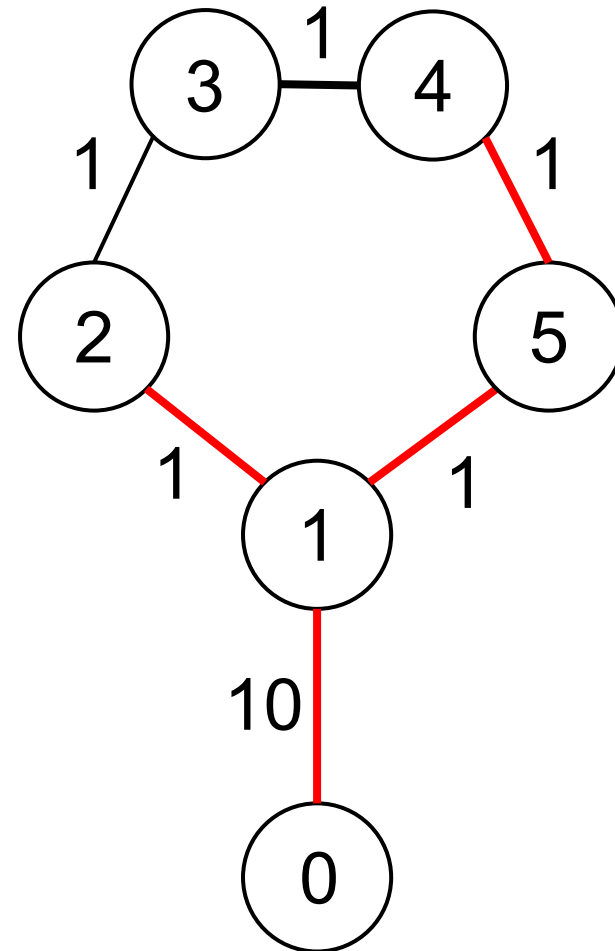
2 3 1

3 4 1

4 5 1

Saída esperada:

14



Roteiro da atividade

Observações sobre a implementação:

- **Em C:** Você deve implementar o TAD UnionFind bem como as funções propostas no arquivo unionFind.h
- **Em C++:** Você deve implementar a classe do arquivo unionFind.hpp.
- Em ambos os casos você não deve alterar a interface das funções ou métodos (você não pode adicionar campos ou métodos a classe, nem modificar as assinaturas dos métodos ou funções), mas pode adicionar os includes de seus TADs.

unionFind.h - Interface do TAD e suas funções

```
typedef struct s_edge{
    int u;
    int v;
    int custo;
} Aresta;

typedef struct s_dsu{
    int representante;
    int rank;
} UnionFind;

UnionFind* NovoUnionFind( int quantidade_subconjuntos);
void DeletaUnionFind(UnionFind* dsu);
void Make(UnionFind* dsu, int x);
int Find(UnionFind* dsu, int x);
void Union(UnionFind* dsu, int x, int y);
```

unionFind.hpp - Interface da classe

```
typedef struct s_edge{
    int u;
    int v;
    int custo;
} Aresta;

typedef struct s_subset{
    int representante;
    int rank;
} Subconjunto;

class UnionFind{
public:
    UnionFind(int quantidade_subconjuntos);
    ~UnionFind();
    void Make(int x);
    int Find(int x);
    void Union(int x, int y);
private:
    int tamanho;
    Subconjunto* subconjuntos;
};
```

Submissão

- A submissão será feita por **VPL**. Certifique-se de seguir as instruções do tutorial disponibilizado no moodle.
 - O seu arquivo executável **DEVE** se chamar **pa4.out** e deve estar localizado na pasta **bin**.
 - Seu código será compilado com o comando:
 make all
 - Você **DEVE** utilizar a estrutura de projeto abaixo junto ao Makefile :
 - PA4
 - |- src
 - |- bin
 - |- obj
 - |- include
- Makefile