

Trabalho de Implementação

Problema. ¹ Rússia 1895. Os idealistas estão espalhados por todo o país. Suas ambições revolucionárias dependem de sua habilidade para fazer e manter contato uns com os outros. Porém, os canais de comunicação podem ser arriscados de estabelecer, utilizar e manter. Os revolucionários precisam estabelecer canais suficientes de comunicação entre todos os pares de indivíduos, de forma que uma mensagem possa ser enviada –talvez indiretamente– de qualquer pessoa para qualquer outra pessoa. Naturalmente eles preferem canais seguros e não mais do que os necessários, isto é: que o risco total seja minimizado. Além disso, eles querem acelerar a revolução estabelecendo a rede o mais rápido possível.

Muitas estratégias foram propostas pelos teóricos. Surpreendentemente todas chegaram à mesma rede. Duas delas chegaram até o teu laboratório no Рио де Јанеиро.

Kruskal “Os canais mais seguros devem ser estabelecidos primeiro e os mais arriscados deixados como último recurso. Os primeiros dois indivíduos a estabelecer contato na Rússia devem ser aqueles que podem fazê-lo da forma mais segura. O par seguinte mais seguro, mesmo esteja na longínqua Sibéria, deve estabelecer contato.”

Prim “Começando por Lenin, a organização revolucionária central em São Petersburgo deve crescer recrutando como novos membros, os amigos mais próximos de um membro atual –tão próximo quanto for possível.”

Objetivo [Em duplas] Implementar e analisar os algoritmos dos camaradas Kruskal e Prim.

Fase 1. Implementar os algoritmos a seguir:

1. Algoritmo de Kruskal utilizando o heap sort e a estrutura union-find (utilizando as heurísticas de union by rank e path compression).
2. Algoritmo de Kruskal utilizando o counting sort e a estrutura union-find (utilizando as heurísticas de union by rank e path compression).
3. Algoritmo de Prim utilizando a fila de prioridade sobre as arestas.
4. Algoritmo de Prim utilizando a fila de prioridade com a operação change-key sobre os vértices.

Fase 2. Executar os algoritmos sobre as instâncias dadas. O formato de entrada é o seguinte:

```
[número de vértices]
[aresta 1, vértice 1] [aresta 1, vértice 2] [risco aresta 1]
[aresta 2, vértice 1] [aresta 2, vértice 2] [risco aresta 2]
...
```

¹Jason Eisner, *State-of-the-Art Algorithms for Minimum Spanning Trees - A Tutorial Discussion*, 1997

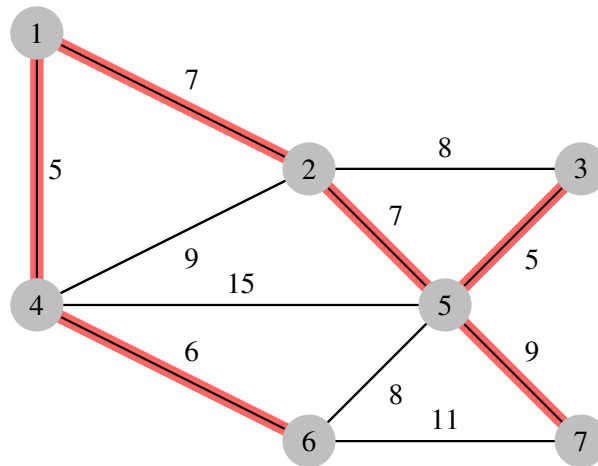


Figure 1: Árvore geradora mínima (arestas ressaltadas) do grafo G

Para toda aresta (u, v) temos que $1 \leq u < v \leq n$, onde n é o número de vértices. Pode assumir que todos os riscos (custos) são inteiros não negativos, mas não são necessariamente distintos.

Por exemplo, a seguir a entrada correspondente ao grafo da Figura 1:

```

7
1 2 7
1 4 5
2 3 8
2 4 9
2 5 7
3 5 5
4 5 15
4 6 6
5 6 8
5 7 9
6 7 11

```

Fase 3. Escrever um relatório descrevendo o trabalho realizado. Este deve incluir:

- Decisões de implementação que foram importantes no desenvolvimento do trabalho;
- O ambiente computacional utilizado;
- Resultados dos experimentos (tabelas/risco total das árvores);
- Uma discussão comparando os resultados obtidos pelos diferentes algoritmos implementados e confrontando os resultados encontrados com aqueles previstos pela teoria.