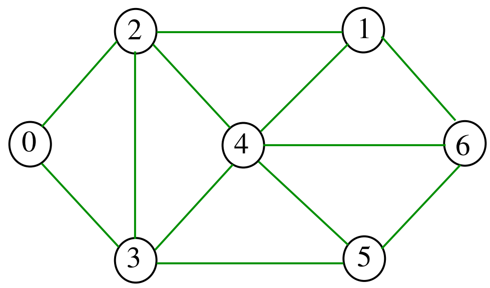
Melhores momentos

AULA 19

Subárvores

Uma **subárvore** de um grafo `G` é qualquer árvore `T` que seja subgrafo de `G`

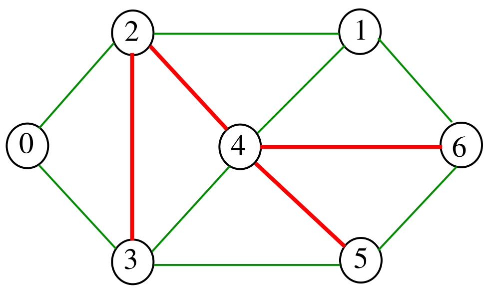
Exemplo:



Subárvores

Uma **subárvore** de um grafo `G` é qualquer árvore `T` que seja subgrafo de `G`

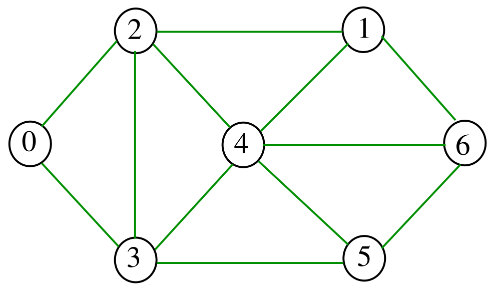
Exemplo: as aretas em vermelho formam uma subárvore



Árvores geradoras

Uma **árvore geradora** (= *spanning tree*) de um grafo é qualquer subárvore que contenha todos os vértices

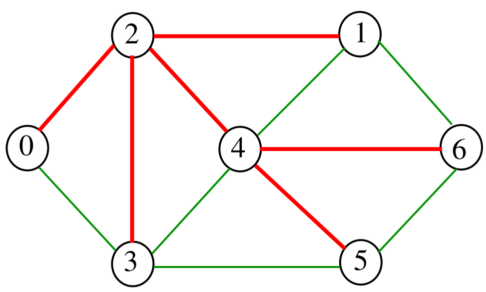
Exemplo:



Árvores geradoras

Uma **árvore geradora** (= *spanning tree*) de um grafo é qualquer subárvore que contenha todos os vértices

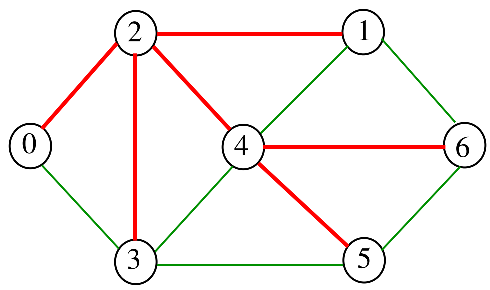
Exemplo: as aretas em vermelho formam uma árvore geradora



Árvores geradoras

Somente grafos conexos têm árvores geradoras
Todo grafo conexo tem uma árvore geradora

Exemplo:



Algoritmos que calculam árvores geradoras

É fácil calcular uma árvore geradora de um grafo conexo:

- a busca em profundidade e
- a busca em largura

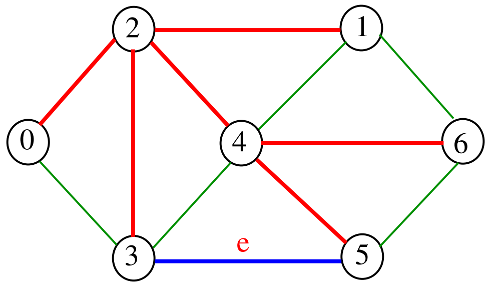
fazem isso.

Qualquer das duas buscas calcula uma arborescência que contém um dos arcos de cada aresta de uma árvore geradora do grafo

Primeira propriedade da troca de arestas

Seja T uma árvore geradora de um grafo G
Para qualquer aresta e de G que não esteja em T, T+e tem um **único ciclo** não-trivial

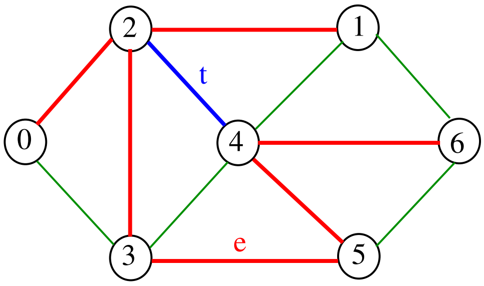
Exemplo: T+e



Primeira propriedade da troca de arestas

Seja T uma árvore geradora de um grafo G
Para qualquer aresta t desse ciclo, T+e-t é uma **árvore geradora**

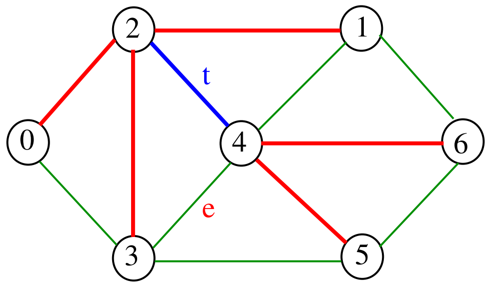
Exemplo: T+e-t



Segunda propriedade da troca de arestas

Seja T uma árvore geradora de um grafo G.
Para qualquer aresta t de T e qualquer aresta e que atravesse o corte determinado por T-t, o grafo T-t+e é uma árvore geradora

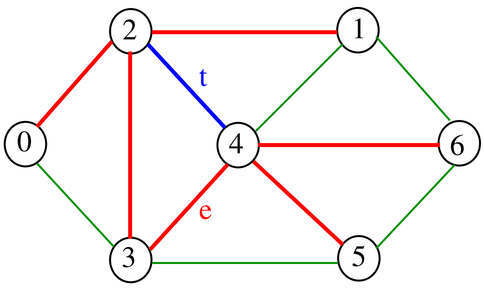
Exemplo: T-t



Segunda propriedade da troca de arestas

Seja T uma árvore geradora de um grafo G.
Para qualquer aresta t de T e qualquer aresta e que atravesse o corte determinado por T-t, o grafo T-t+e é uma árvore geradora

Exemplo: T-t+e



AULA 20

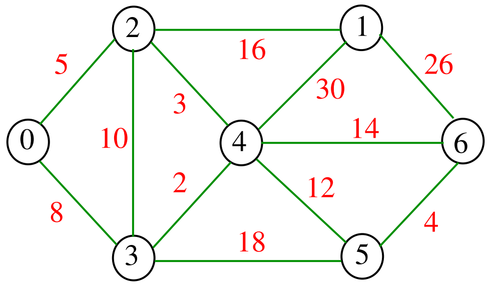
Árvores geradoras de custo mínimo

S 20.1 e 20.2

Árvores geradoras mínimas

Uma **árvore geradora mínima** (= *minimum spanning tree*), ou MST, de um grafo com custos nas arestas é qualquer árvore geradora do grafo que tenha custo mínimo

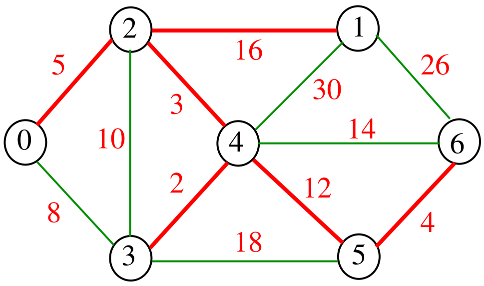
Exemplo: um grafo com custos nas aretas



Árvores geradoras mínimas

Uma **árvore geradora mínima** (= *minimum spanning tree*), ou MST, de um grafo com custos nas arestas é qualquer árvore geradora do grafo que tenha custo mínimo

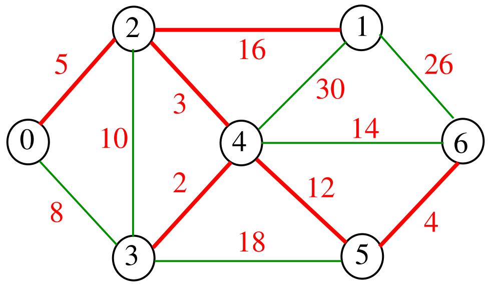
Exemplo: MST de custo 42



Problema MST

Problema: Encontrar uma MST de um grafo `G` com custos nas arestas
O problema tem solução se e somente se o grafo `G` é conexo

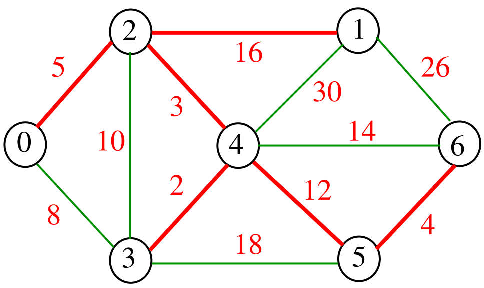
Exemplo: MST de custo 42



Propriedade dos ciclos

Condição de Otimalidade: Se `T` é uma MST então toda aresta `e` fora de `T` tem custo máximo dentre as arestas do único ciclo não-trivial em `T+e`

Exemplo: MST de custo 42



Demonstração da recíproca

Seja `T` uma árvore geradora satisfazendo a condição de otimalidade.

Vamos mostrar que `T` é uma MST.

Demonstração da recíproca

Seja $\mathtt{T}$ uma árvore geradora satisfazendo a condição de otimalidade.

Vamos mostrar que $\mathtt{T}$ é uma MST.

Seja $\mathtt{T}^*$ uma MST tal que o número de arestas comuns entre $\mathtt{T}$ e $\mathtt{T}^*$ seja máximo.

Se $\mathtt{T} = \mathtt{T}^*$ não há o que demonstrar.

Suponha que $\mathtt{T} \neq \mathtt{T}^*$ e seja $\mathtt{e}^*$ uma aresta de custo mínimo dentre as arestas que estão em $\mathtt{T}^*$ mas **não estão** em $\mathtt{T}$.

Seja $\mathtt{e}$ uma aresta qualquer que **está** no único ciclo em $\mathtt{T}+\mathtt{e}^*$ mas **não está** em $\mathtt{T}^*$.

Continuação

Logo, $\mathtt{custo}(\mathtt{e}) \leq \mathtt{custo}(\mathtt{e}^*)$.

Seja $\mathtt{f}^*$ uma aresta qualquer que **está** no único ciclo em $\mathtt{T}^*+\mathtt{e}$ mas **não está** em $\mathtt{T}$.

Como $\mathtt{T}^*$ é uma MST então $\mathtt{custo}(\mathtt{f}^*) \leq \mathtt{custo}(\mathtt{e})$.
Se $\mathtt{custo}(\mathtt{f}^*) = \mathtt{custo}(\mathtt{e})$, então $\mathtt{T}^*-\mathtt{f}^*+\mathtt{e}$ é uma MST que contradiz a escolha de $\mathtt{T}^*$. Logo, $\mathtt{custo}(\mathtt{f}^*) < \mathtt{custo}(\mathtt{e})$.
Finalmente, concluímos que

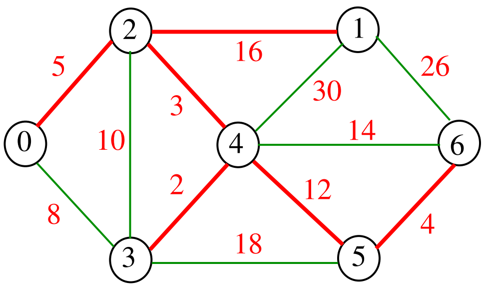
$$\mathtt{custo}(\mathtt{f}^*) < \mathtt{custo}(\mathtt{e}) \leq \mathtt{custo}(\mathtt{e}^*)$$

o que contradiz a nossa escolha de $\mathtt{e}^*$.
Portanto, $\mathtt{T} = \mathtt{T}^*$ e $\mathtt{T}$ é uma MST.

Propriedade dos cortes

Condição de Otimalidade: Se $\mathtt{T}$ é uma MST então cada aresta $\mathtt{t}$ de $\mathtt{T}$ é uma aresta mínima dentre as que atravessam o corte determinado por $\mathtt{T}$-$\mathtt{t}$

Exemplo: MST de custo 42



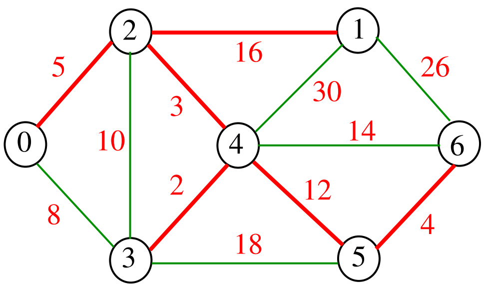
Algoritmo de Prim

S 20.3

Problema MST

Problema: Encontrar uma MST de um grafo $\mathtt{G}$ com custos nas arestas
O problema tem solução se e somente se o grafo $\mathtt{G}$ é conexo

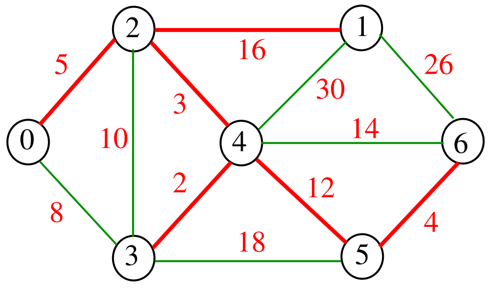
Exemplo: MST de custo 42



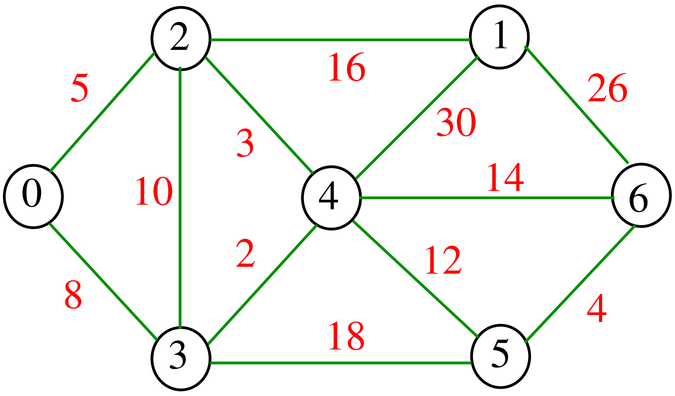
Propriedade dos cortes

Condição de Otimalidade: Se $\mathtt{T}$ é uma MST então cada aresta $\mathtt{t}$ de $\mathtt{T}$ é uma aresta mínima dentre as que atravessam o corte determinado por $\mathtt{T}$-$\mathtt{t}$

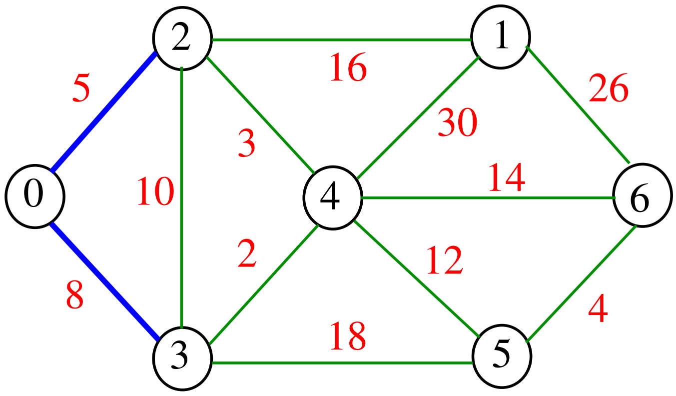
Exemplo: MST de custo 42



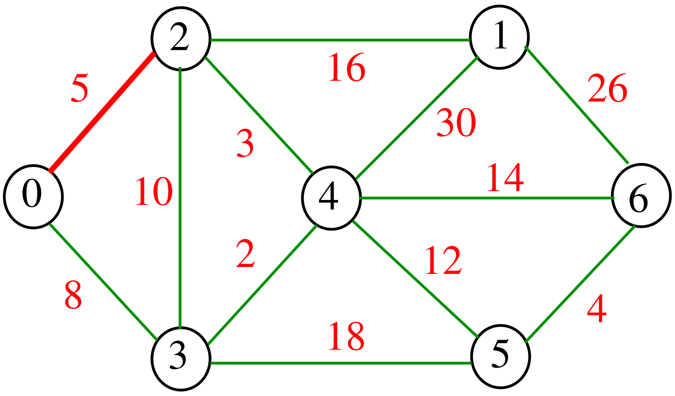
Simulação



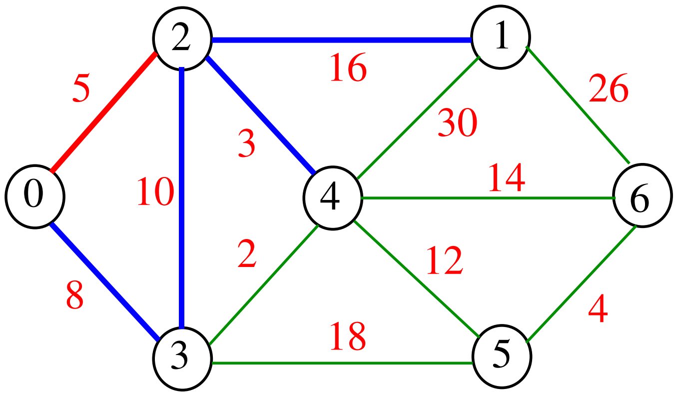
Simulação



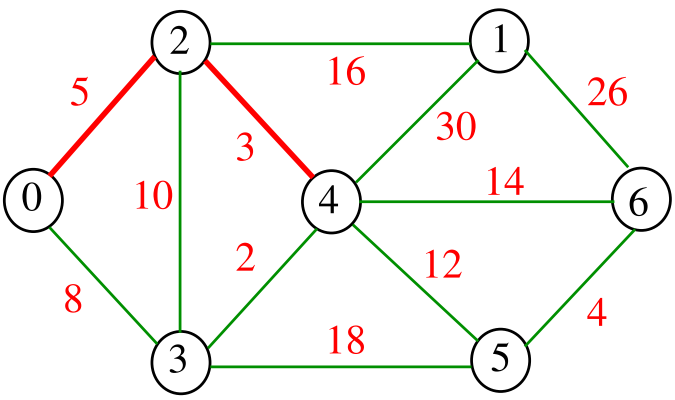
Simulação



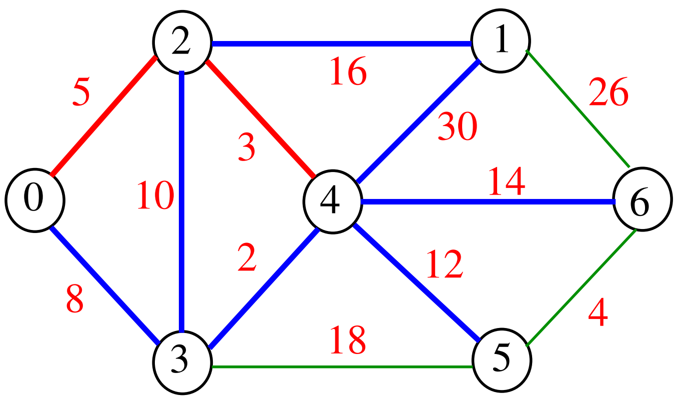
Simulação



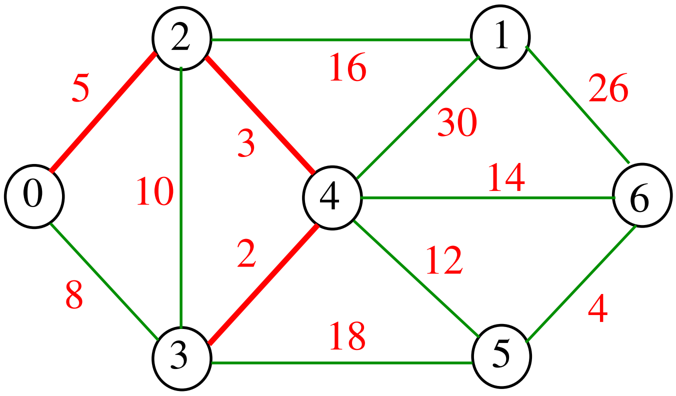
Simulação



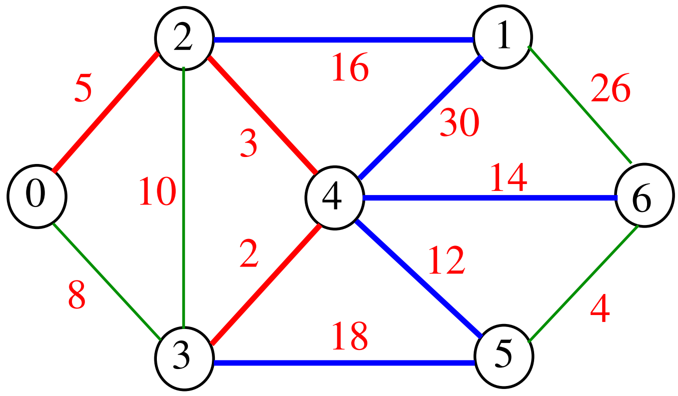
Simulação



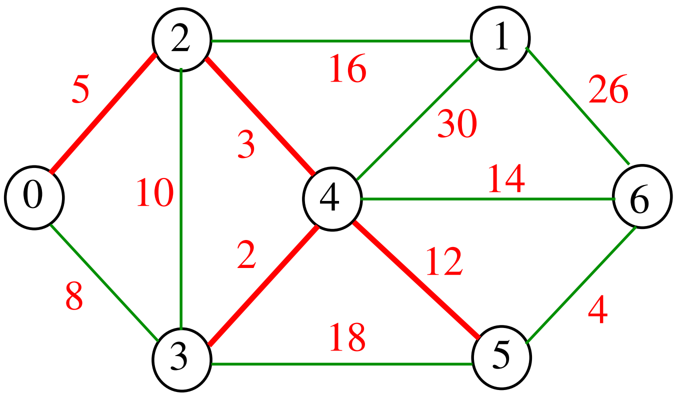
Simulação



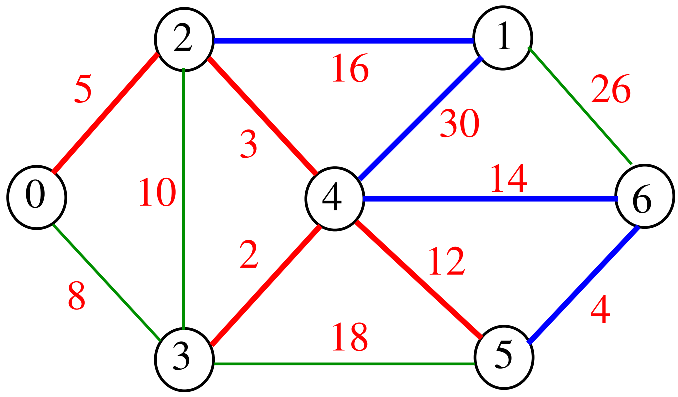
Simulação



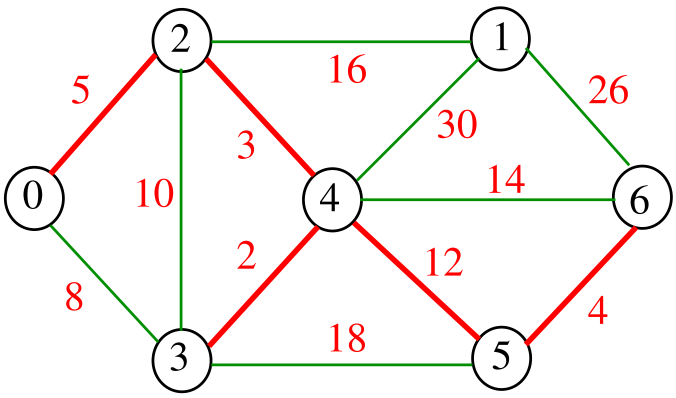
Simulação



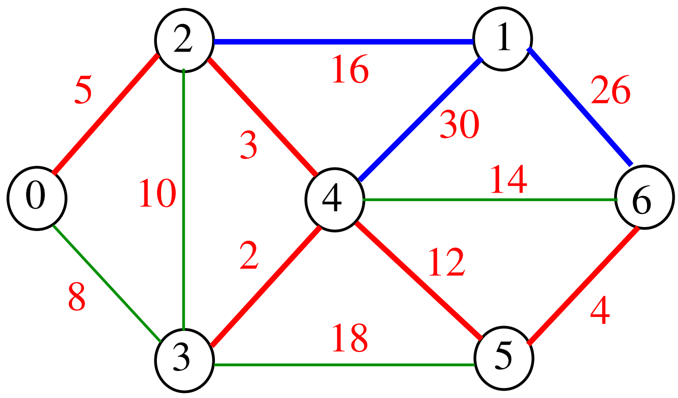
Simulação



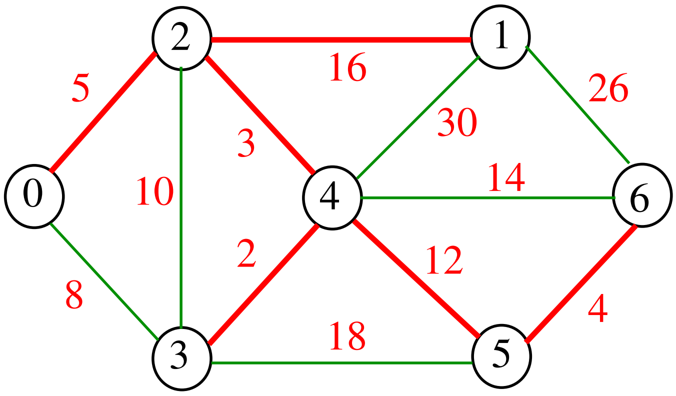
Simulação



Simulação



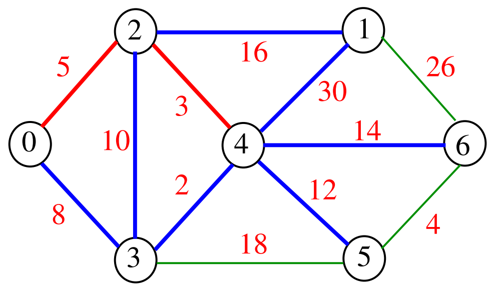
Simulação



Franja

A **franja** (= *fringe*) de uma subárvore T é o conjunto de todas as arestas que têm uma ponta em T e outra ponta fora

Exemplo: As arestas em azul formam a franja de T



Algoritmo de Prim

O algoritmo de Prim é iterativo.

Cada iteração começa com uma subárvore T de G.

No início da primeira iteração T é um árvore com apenas 1 vértice.

Cada iteração consiste em:

Caso 1: franja de T é vazia
Devolva T e pare.

Caso 2: franja de T não é vazia
Seja e uma aresta de custo mínimo na franja de T
Comece nova iteração com T+e no papel de T

Relação invariante chave

No início de cada iteração vale que

existe uma MST que contém as arestas em T.

Se a relação vale no **início da última** iteração então é evidente que, se o grafo é conexo, o algoritmo devolve uma **MST**.

Demonstração. Vamos mostrar que se a relação vale no início de uma iteração que não seja a última, então ela vale no fim da iteração com T+e no papel de T.

A relação invariante certamente vale no início da primeira iteração.

Demonstração

Considere o início de uma iteração qualquer que não seja a última.

Seja e a aresta escolhida pela iteração no caso 2.

Pela relação invariante existe uma MST T^* que contém T.

Se e está em T^*, então não há o que demonstrar. Suponha, portanto, que e não está em T^*.

Seja e^* uma aresta que está no único ciclo em T^*+e que está na franja de T.

Pela escolha de e temos que $\texttt{custo}(e) \leq \texttt{custo}(e^*)$.

Portanto, T^*-e^*+e é uma MST que contém T+e.