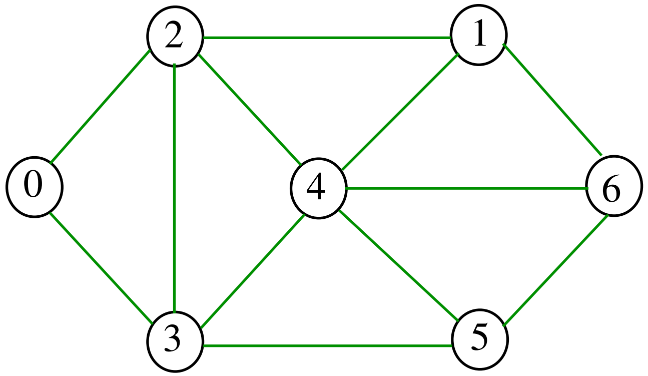
Árvores geradoras de grafos

Árvores geradoras

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

Exemplo:

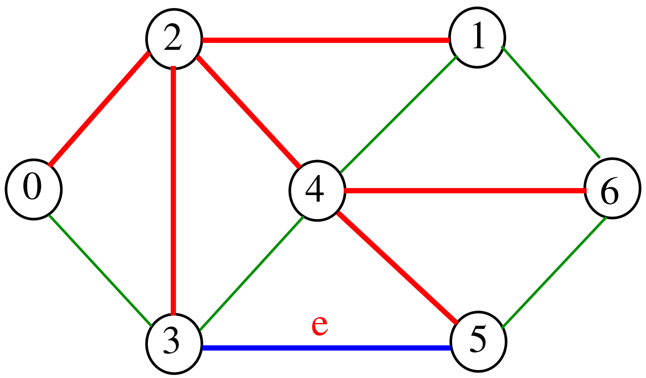


Exemplo: as arestas em vermelho formam uma árvore

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, o **ciclo fundamental** $C(\mathtt{T}, \mathtt{e})$.

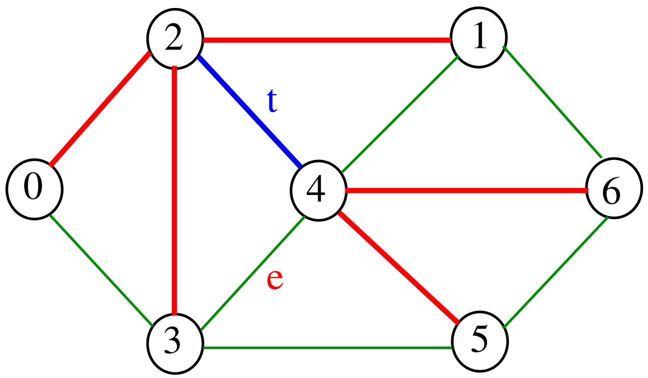
Exemplo: T+e



Segunda propriedade da troca de arestas

Seja T uma árvore geradora de um grafo G
Para qualquer aresta t de T, T-t tem duas componentes. O corte em G que separa essas componentes é o **corte fundamental** $D(\mathtt{T}, \mathtt{t})$.

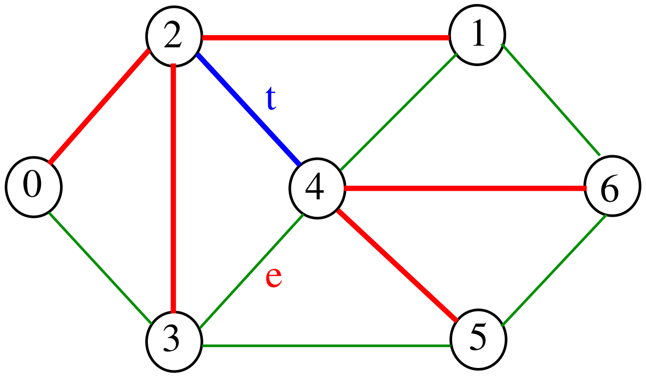
Exemplo: T-t $\qquad \{0,1,2,3\} \xrightarrow{D(\mathtt{T},\mathtt{t})} \{4,5,6\}$



Segunda propriedade da troca de arestas

Seja T uma árvore geradora de um grafo G
Para qualquer aresta t de T, se $e \in D(T, t)$ então
$T - t + e$ é uma árvore geradora.

Exemplo: T-t



Exemplo: T-t+e

Propriedade fundamental

Se T é árvore geradora de G, t é uma aresta de T e e é uma aresta fora de T, então

$$t \in C(T, e) \iff e \in D(T, t).$$

Dem: São equivalentes

- $t \in C(T, e)$
- t está no caminho em T que liga as pontas de e
- $T - t$ não tem caminho entre as pontas de e
- $e \in D(T, t)$.

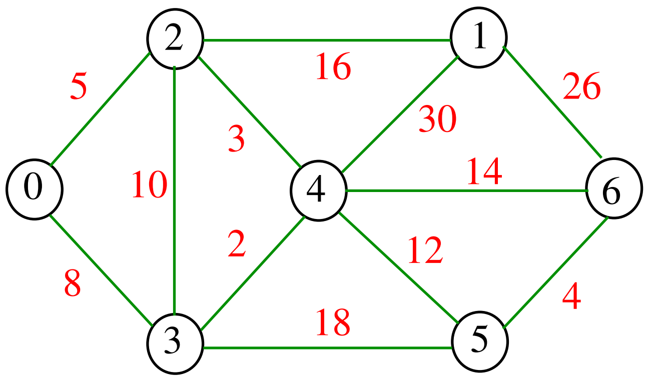
Árvores geradoras de custo mínimo

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 arestas

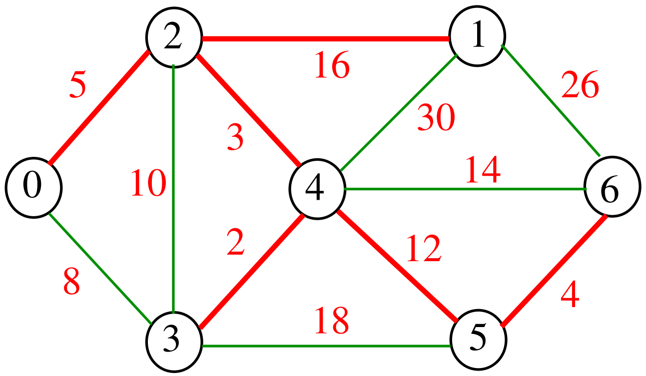


Problema MST

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

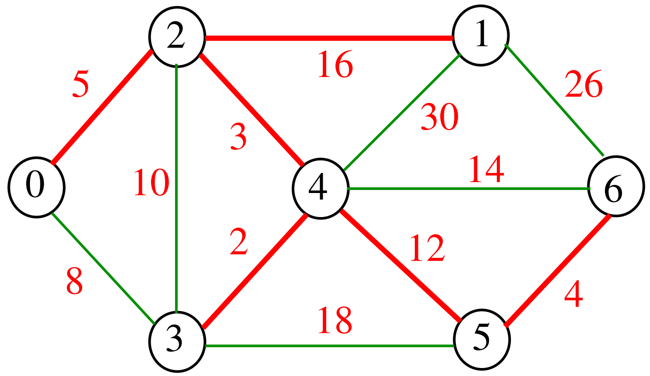
Exemplo: MST de custo 42



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 cilo fundamental 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.

Seja M uma MST tal que o número de arestas comuns entre T e M seja máximo.

Se $T = M$ não há o que demonstrar.

Suponha que $T \neq M$ e seja e uma aresta de custo mínimo dentre as arestas que estão em M mas não estão em T.

Seja d uma aresta qualquer que **não está** em M mas **está** no ciclo fundamental $C(T, e)$.

Continuação

Logo, $\mathtt{custo}(d) \leq \mathtt{custo}(e)$ (1).

Seja f uma aresta qualquer em $C(M, d) - T$.

Como M é uma MST, $\mathtt{custo}(f) \leq \mathtt{custo}(d)$ (2).

Pela escolha de e, $\mathtt{custo}(e) \leq \mathtt{custo}(f)$ (3).

Juntando (1), (2) e (3), vem que

$$\mathtt{custo}(d) = \mathtt{custo}(f) = \mathtt{custo}(e)$$

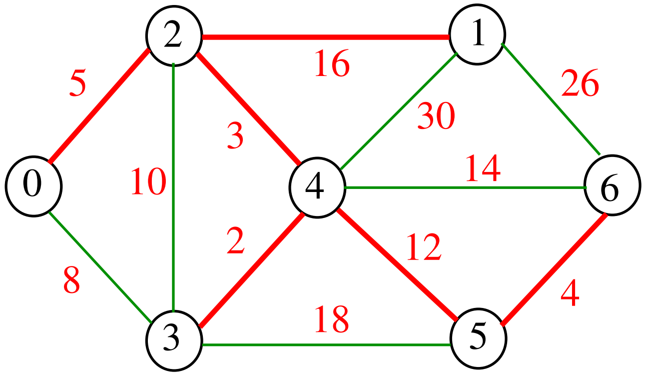
Mas então, $M - f + d$ é uma MST que tem o mesmo custo que M, logo é mínima. Por outro lado, tem uma aresta a mais em comum com T do que M. Isso contradiz a escolha de M.

Portanto, $T = M$, o que mostra que T é uma MST.

Propriedade dos cortes

Condição de Otimalidade: T é uma MST se e somente se cada aresta t de T é uma aresta mínima no corte fundamental T, t.

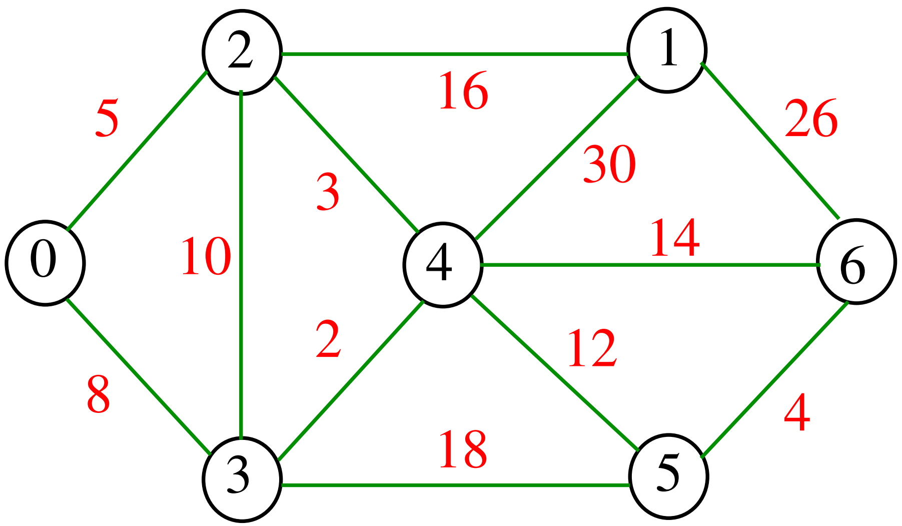
Exemplo: MST de custo 42



Algoritmo de Prim

S 20.3

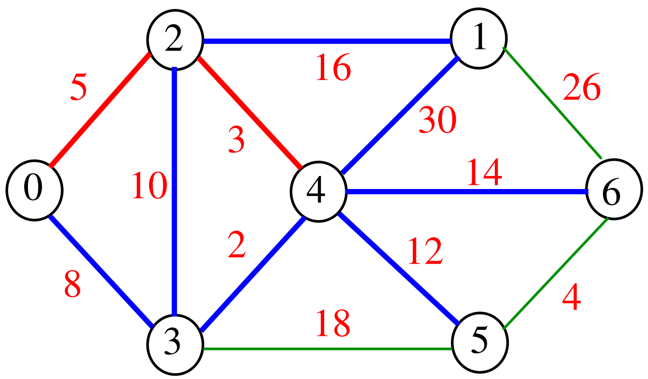
Simulação



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`
Faça `T` $\leftarrow$ `T` + `e`

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 `M` que contém `T`.

Se `e` está em `M`, então não há o que demonstrar. Suponha, portanto, que `e` não está em `M`.

Seja `t` uma aresta que está C(`M`, `e`) que está na franja de `T`. Pela escolha de `e` feita pelo algoritmo, `custo(e)` $\leq$ `custo(t)`.

Portanto, `M−t+e` é uma MST que contém `T+e`.