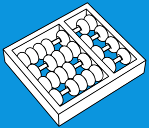
Árvore geradora mínima

MC558 - Projeto e Análise de Algoritmos II

08/24

08



Santiago Valdés Ravelo
https://ic.unicamp.br/~santiago/
ravelo@unicamp.br

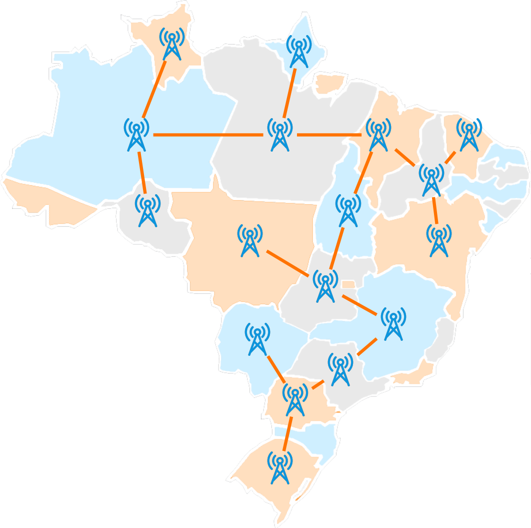
UNICAMP

"Árvores Geradoras Mínimas... ou como unir o mundo com um orçamento limitado."

Seth James Nielson

Árvore geradora mínima

Motivação



Motivação

Na situação anterior:

- Temos um conjunto de torres.
- Para conectar duas torres usamos um cabeamento.
- Queremos que todas elas estejam interconectadas.

Como **MINIMIZAR** o comprimento do cabeamento?

- Este é um problema de otimização.
- A entrada pode ser modelada como um grafo

Árvore geradora mínima

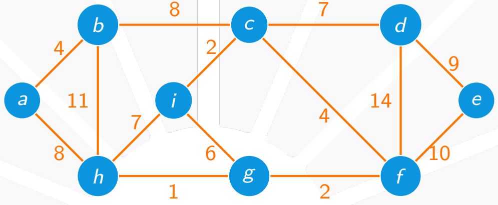
Problema (Árvore geradora mínima (AGM))

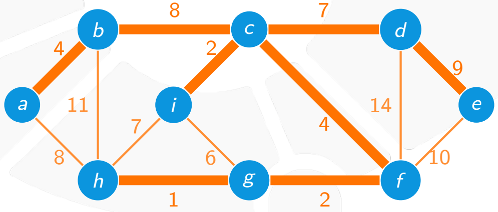
Entrada: *Um grafo conexo* $G = (\boldsymbol{V}, \boldsymbol{E})$ *com peso* $w(\boldsymbol{u,v}) \geq 0$ *para cada aresta* ***(u,v)***.

Solução: *Um subgrafo gerador conexo* T *de* G.

Objetivo: ***MINIMIZAR***: $w(T) = \sum_{(u,v) \in E[T]} w(\boldsymbol{u,v})$.

Exemplo


b
c
d
a
i
e
h
g
f
8
7
4
2
9
11
14
4
7
8
6
10
1
2


b
c
d
a
i
e
h
g
f
8
7
4
2
9
11
14
4
7
8
6
10
1
2

Refletindo um pouco

Observações:

- Se o grafo fosse desconexo, então não haveria solução.
- Supomos que **NÃO** há arestas de peso negativo.

Algumas perguntas:

- Por que dizemos que uma solução ótima é uma **ÁRVORE**?
- Vale se houvesse arestas de peso negativo na entrada?

Algoritmos estudados

Veremos dois algoritmos **GULOSOS**:

1. Algoritmo de Prim.
2. Algoritmo de Kruskal.

Esquema dos algoritmos

Ideia:

- Construiremos uma árvore incrementalmente.
- Denotamos por **A** um conjunto de arestas da árvore.
- Garantimos que **A** seja um subconjunto de uma AGM.
- Mantemos essa invariante em cada iteração:
  1. No início da iteração, **A** satisfaz a invariante.
  2. Selecionamos **(u,v)** tal que $\mathbf{A} \cup \{\mathbf{(u,v)}\}$ também a satisfaça.
  3. Adicionamos **(u,v)** ao conjunto **A**.

Dizemos que uma tal **(u,v)** é uma **ARESTA SEGURA**.

Algoritmo genérico

Algoritmo: $\textsc{AGM-Generico}(G, w)$

$A \leftarrow \emptyset$
enquanto *A não é uma árvore geradora*
    encontre uma aresta segura (u, v) para A
    $A \leftarrow A \cup \{(u, v)\}$
devolva A

O algoritmo está correto:

- O algoritmo devolve uma árvore geradora **A**.
- **A** é subgrafo de alguma AGM.
- Logo, **A** é também mínima.

Algoritmo genérico

O algoritmo está bem definido?

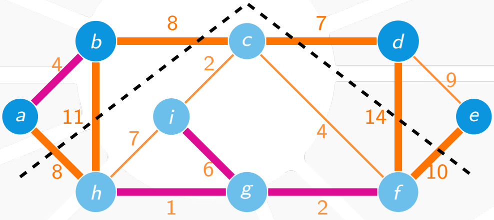
- Se a iteração executa, então **A** não é árvore geradora.
- Logo, **A** não contém todas arestas de alguma AGM T.
- Assim qualquer aresta de **E[T]** $\setminus$ **A** é segura.

Os algoritmos que veremos diferem em como encontrar uma **ARESTA SEGURA**.

Como encontrar arestas seguras?

Considere um grafo $G = (\mathbf{V}, \mathbf{E})$ e seja $\mathbf{S} \subset \mathbf{V}$.

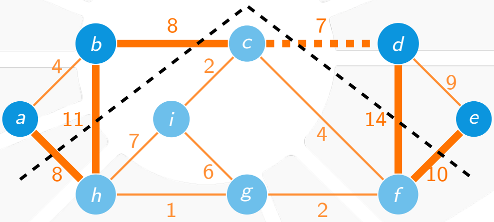
- Denote por $\delta(\mathbf{S})$ o conjunto de arestas de G com um extremo em $\mathbf{S}$ e outro em $\mathbf{V} \setminus \mathbf{S}$.
- Lembre-se de que um tal conjunto é chamado de **CORTE**.



Um corte $\delta(\mathbf{S})$ **RESPEITA** um conjunto $\mathbf{A}$ de arestas se não contiver nenhuma aresta de $\mathbf{A}$.

Arestas leves

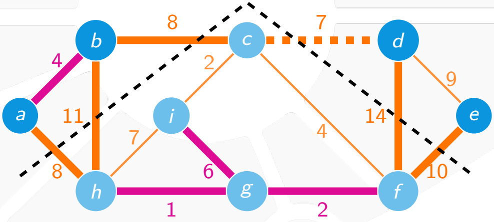
Uma aresta de um corte $\delta(S)$ é **LEVE** se tem o menor peso entre as arestas do corte.



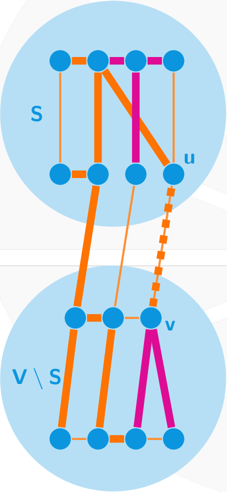
Arestas leves

Teorema

Seja (G, w) um grafo com pesos nas arestas e suponha que ***A*** *é um subconjunto de arestas de uma AGM de G. Se $\delta(\mathbf{S})$ é um corte que respeita* ***A*** *e* ***(u,v)*** *é uma aresta leve desse corte, então* ***(u,v)*** *é uma* ***ARESTA SEGURA****.*



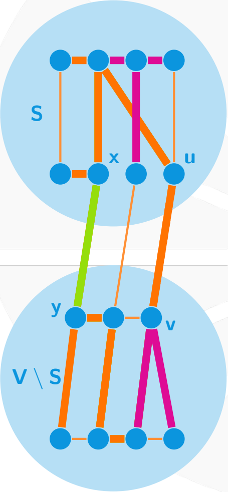
Demonstração do teorema



Seja **T** uma AGM que contém **A**:

- Tome $\delta(\mathbf{S})$ um corte que respeita **A**.
- Seja **(u,v)** uma **ARESTA LEVE** deste corte.
- Se **(u,v)** estiver em T, então não há nada a mostrar.
- Se **(u,v)** **NÃO** for uma aresta de T:
  - Construiremos uma AGM T' que contém $\mathbf{A} \cup \{\mathbf{(u,v)}\}$,
  - concluindo que **(u,v)** é **SEGURA**.

Demonstração do teorema



Suponha que **(u,v)** **NÃO** for uma aresta de T:

- Existe um único caminho **P** de **u** a **v** em T.
- **u** a **v** estão em lados opostos do corte $\delta(\mathbf{S})$.
- Então, alguma aresta de **P** pertence ao corte.
- Seja **(x,y)** uma tal aresta (note que **(x,y)** $\notin$ **A** pois o corte respeita **A**).
- Defina $T' = T - \mathbf{(x,y)} + \mathbf{(u,v)}$.
- Observe que T' é uma árvore geradora.
- Como **(u,v)** é uma **ARESTA LEVE** de $\delta(\mathbf{S})$, temos que $w(\mathbf{u,v}) \leq w(\mathbf{x,y})$.
- Logo, $w(T') = w(T) - w(\mathbf{x,y}) + w(\mathbf{u,v}) \leq w(T)$.
- Portanto, T' é uma **AGM**.

Consequência para os algoritmos

Corolário

Seja (G, w) um grafo com pesos nas arestas e suponha que $\boldsymbol{A}$ *é um subconjunto de arestas de uma AGM de G. Se* $\boldsymbol{C}$ *são os vértices de uma componente de $G_A = (\boldsymbol{V}, \boldsymbol{A})$ e* ***(u,v)*** *é uma aresta leve do corte $\delta(\boldsymbol{C})$, então* ***(u,v)*** *é uma* ***ARESTA SEGURA****.*

Isso sugere uma estratégia iterativa:

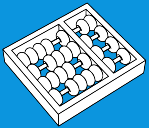
- Prim e Kruskal implementam essa ideia.
- Tais algoritmos farão uso desse corolário.

Árvore geradora mínima

MC558 - Projeto e Análise de Algoritmos II

08/24

08



Santiago Valdés Ravelo
https://ic.unicamp.br/~santiago/
ravelo@unicamp.br

UNICAMP