

# 12 - AGM - Prim - Kruskal

## Árvore geradora mínima - lembrete:

Em um grafo não direcionado é um subgrafo gerador que também é uma árvore.



## Árvore geradora de custo mínimo:

Seja um grafo não direcionado e ponderado em que o valor do custo  $> 0$  é associado a cada aresta. O custo total de uma árvore geradora pode ser obtido pela soma dos custos de todas as arestas do grafo. Então a árvore geradora de custo mínimo é a árvore geradora de menor custo total dentre todas as possíveis árvores geradoras de um grafo.

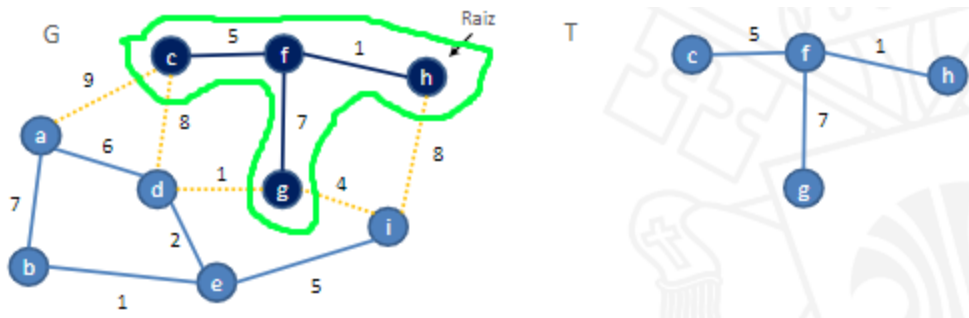
## Método de Prim:

Incluir de forma gulosa, um a um, os vértices da árvore geradora mínima. A partir de um vértice qualquer (raiz), a cada passo, acrescenta-se a aresta de menor peso incidente ao conjunto de vértices que já foram selecionado e que possui uma extremidade em vértices no conjunto de não selecionados.

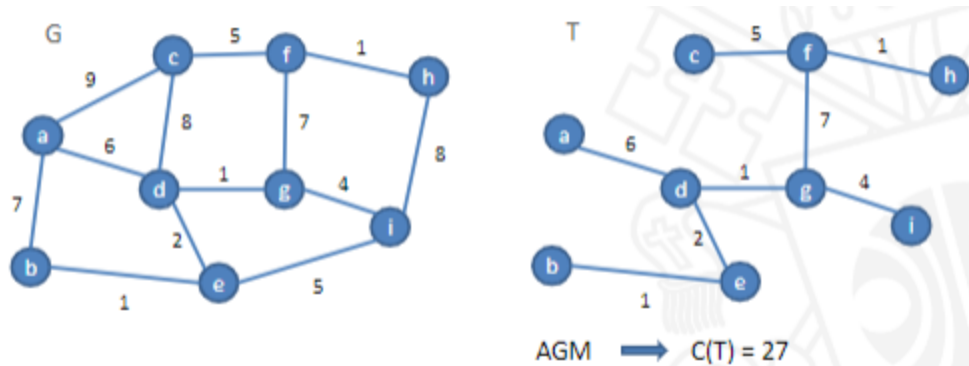
## Algoritmo:

1. Escolher um vértice qualquer  $r \in V(G)$  // Selecionar uma raiz
2.  $V(T) \leftarrow \{r\};$  // Inicializar conj. de vértices selecionados
3.  $E(T) \leftarrow \emptyset;$  // Inicializar conj. arestas da AGM
4. enquanto  $V(T) \neq V(G)$  efetuar // Se houver vértice não selecionado?
  - a. Encontrar a aresta  $\{v, w\}$  de menor custo tal que  $v \in V(T)$  e  $w \notin V(T)$   
// Isto é, a aresta de menor custo entre selecionados e não selecionados
  - b. Acrescentar  $w$  a  $V(T)$  // Adicionar novo vértice à AGM
  - c. Acrescentar  $\{v, w\}$  a  $E(T)$  // Adicionar nova aresta à AGM

Como fazer, para cada conjunto de vértices você vai olhar a aresta incidentes a todos aqueles, para que você sempre escolha aquela aresta com menor custo entre todas as arestas e também deve-se lembrar que a árvore geradora não pode fazer um ciclo.



Pegar a aresta de menor custo incidente aos vértices que estão dentro do “componente” verde, caso a aresta descubra um novo vértice adicionar o vértice no “componente” e continuar.



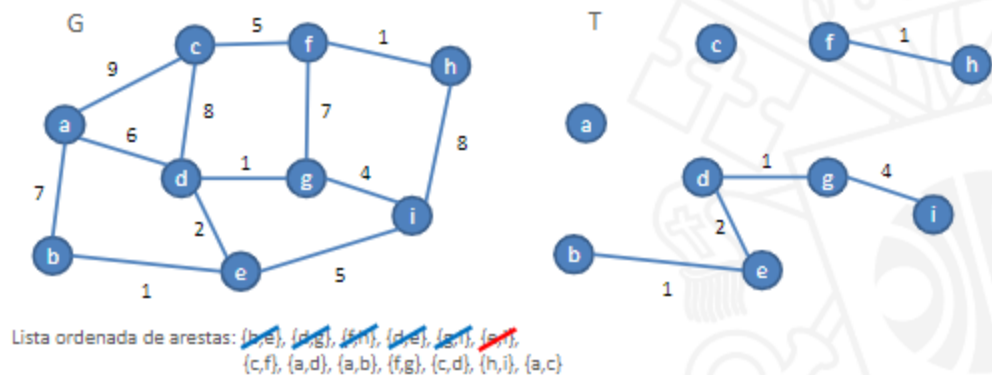
## Método de Kruskal:

Incluir na árvore a cada interação a aresta de menor custo que não forme um ciclo com as inseridas anteriormente. Dessa forma são necessárias pelo menos  $n - 1$  interações. Contudo, pode ser necessário fazer mais interações devido a necessidade de ignorar arestas que formem ciclos.

Algoritmo:

1. Ordenar arestas em ordem não decrescente  $e_1, e_2, e_3, \dots$
2.  $V(T) \leftarrow V(G)$ ; // Adiciona todos vértices à AGM
3.  $E(T) \leftarrow \{ e_1 \}$ ; // Inicializar arestas da AGM
4.  $j \leftarrow 2$ ; // Indica aresta a ser analisada
5. enquanto  $|E(T)| < |V(T)| - 1$  efetuar // Se ainda não for conexo?
  - a. se aresta  $e_j$  não forma ciclo com as arestas em  $E(T)$  então
    - i. Acrescentar  $\{v, w\}$  a  $E(T)$  // Adicionar nova aresta à AGM
  - b.  $j \leftarrow j + 1$ ; // Indica próxima aresta

Para fazer pelo método de Kruskal basta ter todos os vértices sem nenhuma aresta, criar uma lista ordenada de arestas pelo seu custo e ir adicionando na ordem da lista, caso alguma aresta forme um ciclo, pula e não adiciona ela.



Pode-se observar que a aresta  $\{e, i\}$  não pode ser adicionada, pois irá formar um ciclo caso isso aconteça.

