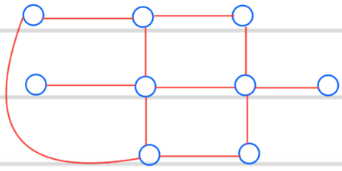
27/03/2025

$G = (V, E)$, simples, não-direcionado, conexo



- Projete uma solução para remover arestas de forma a preservar a conectividade.
  - Conjunto disjuntos ou Union-find (outra estrutura)
  - DFS, se achar ciclo remover essa aresta.
  - BFS, se o vizinho de um vértice já foi visitado ignora/remove a aresta.

→ Árvore Geradora = mesmo conjunto de vértices e um subconjunto de arestas que preserva conectividade.

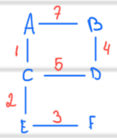
Se não for conexo, mesma coisa para cada componente, assim de árvore passa a virar floresta.

—— // —— // —— // —— // —— // —— // —— // —— // —— // —— //

31/03

- Seja $G = (V, E)$ um grafo não-direcionado e (G, w) um grafo ponderado em que $w: E \to \mathbb{Z}^+$

Projete uma solução para encontrar uma árvore geradora que possua a menor soma de pesos possíveis.

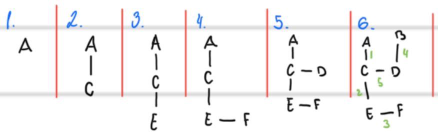


Pegar aresta de menor peso e verificar se fecha ciclo, se fechar, remover a aresta que fecha o ciclo

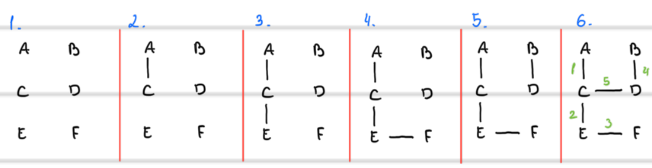
Prim: 1. Parte de um vértice
2. Pego o menor
3. Expande a fronteira

Kruskal: 1. Ordena por peso
2. Seleciona a de menor peso
3. Se não fecha ciclo
1. incluir aresta na árvore

Prim:



Kruskal:



—— // —— // —— // —— // —— // —— // —— // —— // —— // —— //

02/04/2025

- Seja $G = (V, E)$ um grafo não-direcionado e simples, seja (G, w) um grafo ponderado em que $w: E \to \mathbb{Z}^+$.

Projete uma solução para achar a floresta geradora mínima.

↳ Para cada componente conexo aplicar os algoritmos Kruskal ou Prim.

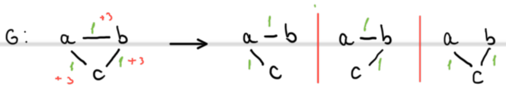
> **Ponderado**
> É um tipo de grafo em que cada aresta possui um valor associado (peso).

- Seja $G = (V, E)$ um grafo não-direcionado, simples e conexo. Seja (G, W) um grafo ponderado em que $w: E \to \mathbb{Z}^+$

A aplicação de um algoritmo que encontra uma árvore geradora mínima produz um grafo $T = (V, E')$.

Se $w'(e) = w(e) + k, \forall e \in E$ em que $k \in \mathbb{Z}^+$ então a árvore geradora mínima T' de (G, w') é igual a T?

↳ G:



↳ NÃO!

> **Principal Diferença**
> Árvore Geradora = Vai gerar qualquer caminho
>
> Árvore Geradora Mínima = Vai gerar o menor caminho

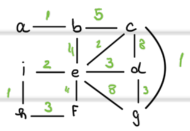
> AGM = MST
> ↳ MINIMUM Spanning Tree
> ↳ ORDENAÇÃO

- $|E| * K$

- Podemos afirmar que AGM conterá a aresta de menor peso do grafo?

↳ Depende, se todos os pesos são diferentes, se todos os pesos são iguais quando tem ciclo.

- G:



Construir AGM com Kruskal e Prim.

—— // —— // —— // —— // —— // —— // —— // —— // —— // —— //

03/04/2025

- Seja $G = (V, E)$ um grafo não-direcionado e simples. Seja (G, W) um grafo ponderado em que $w: E \to \mathbb{Z}^+$.

Projete uma solução para encontrar "dois clusters" em G.

↳ Componentes Conexos.

1. Identificar o menor conjunto de Arestas que disconectam o grafo. (cut set)

2. Usar AGM para remover ciclos e as arestas de maior peso

↳ Eficient Graph-Based Segmentation – Artigo.

↳ Reverse-Delete → AGM Inversa

- Árvore Geradora Mínima

↳ Multiplica os pesos por -1

↳ Usa o algoritmo de Kruskal

—— // —— // —— // —— // —— // —— // —— // —— // —— // —— //

07/04/2025

- Seja $G = (V, E)$ um grafo não-direcionado. Seja (G, W) um grafo ponderado nas arestas. Seja $e \in E$ uma aresta qualquer de E.

  Encontre uma "AGM" de (G, W) e force e estar presente nessa árvore

↳ Forçar uma aresta não necessariamente implica em uma AGM

↳

$p =$ acharMenor(G, W) $W(e) = p - 1$ Kruskal(G, W)	Achar o menor peso do grafo, colocar na aresta que você quer colocar o menor peso, aplicar o Kruskal

- Usar Método: 1. Entender o problema

  2. "Modelar"

     ↳ $G = (V, E) - (G, W)$

  3. "Adaptar" para algo pronto

- Seja $E' \subseteq E$ um subconjunto de arestas E.

  ↳ O grafo terá todas as arestas forçadas mais (+) as arestas da AGM

  ↳ Se $G'E'$ enduzida por E' tiver dica $E'' \subseteq E'$
  incluir $E' \setminus E''$

- P_{xy} é um caminho de x para y
  CAMINHOS

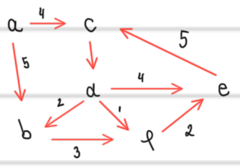
  ↳ $P_{xy} = \{ P(x, y) \mid \exists$ caminho x para $y \}$

  $$W(P_{xy}) = \arg\min_{P \in P_{xy}} \{ W(P) \}$$

  $$W(P) = \sum_{e \in P} w(e)$$

———— // ———— // ———— // ———— // ———— // ———— // ———— // ———— // ———— // ———— //

09/04/2025



- Seja $G = (V, E)$ um grafo direcionado, seja (G, W) um grafo ponderado em que $W: E \to \mathbb{R}^+$

  Projete uma solução para encontrar o menor caminho de "a" para "e".

  ↳ Fazer uma BFS, pegar o caminho com menor soma de peso das arestas entre todas.

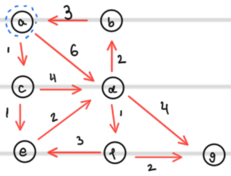
  1. Seleciona o vértice de menor distância acumulada para a origem.

  2. Atualiza vizinhos do vértice escolhido.

  3. Remove o vértice escolhido das possibilidades.

———— // ———— // ———— // ———— // ———— // ———— // ———— // ———— // ———— // ———— //

10/04/2025



- Faça passo-a-passo o "menor caminho" de "a" para todos os vértices

	a	b	c	d	e	f	g
1 -	0	∞	∞	∞	∞	∞	∞
2 -	0	∞	1	∞	∞	∞	∞
3 -	0	∞	1	∞	2	∞	∞
4 -	0	∞	1	4	2	∞	∞
5 -	0	∞	1	4	2	5	∞
6 -	0	∞	1	4	2	5	7
7 -	0	6	1	4	2	5	7

Algoritmo Dijkstra

$S' = V$ // V ainda não visitado

$d[u] = \emptyset$

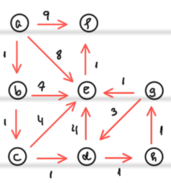
While $S' \neq \emptyset$

v = extrair min de S'

for each $v' \in N(v) \cap S'$

$d[v'] = \min(d[v], d[v] + W(v, v))$

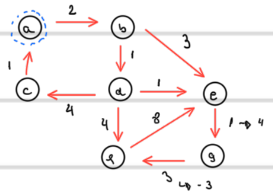
$[n!]$



a	b	c	d	e	f	g	h
∞	∞	∞	∞	∞	∞	∞	∞
0	∞	∞	∞	∞	∞	∞	∞
0	1	∞	∞	8	9	∞	∞
0	1	2	∞	7	9	∞	∞
0	1	2	3	6	9	∞	∞
0	1	2	3	6	9	∞	1
0	1	2	3	6	9	1	1
0	1	2	3	6	9	1	1
0	1	2	3	6	7	1	1

23/04/2025

- Ache o menor caminho de a para todos:



	A	B	C	D	E	F	G
-	∞	∞	∞	∞	∞	∞	∞
0	0	∞	∞	∞	∞	∞	∞
1	0	2	∞	∞	∞	∞	∞
2	0	2	∞	3	5	∞	∞
3	0	2	7	3	4	7	∞
4	0	2	7	3	4	7	5

	~~A~~	~~B~~	~~C~~	~~D~~	~~E~~	~~F~~	~~G~~
-	∞	∞	∞	∞	∞	∞	∞
0	0	∞	∞	∞	∞	∞	∞
1	–	2	∞	∞	∞	∞	∞
2	–	–	∞	3	5	∞	∞
3	–	–	7	–	4	7	∞
4	–	–	7	–	–	7	8
5	–	–	7	–	–	–	8
6	–	–	–	–	–	–	8
7	–	–	–	–	–	–	–

Algoritmo:

for each $v \in V$, $d[v] = \infty$

$S = V$; $d[v] = \emptyset$

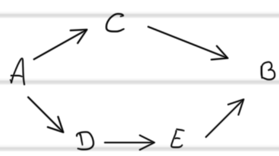
While $S \neq \emptyset$:

  $v^+ = \text{argmin} \{ d[u] \mid u \in S \}$

  $S \leftarrow S \setminus \{v^+\}$

  for each $u \in N(v^+) \cap S$:

    $d[u] = \min(d[u], d[v^+] + w(v^+, u))$



—— // —— // —— // —— // —— // —— // —— // —— // —— // —— //

24/03/2025

- Seja $G = (V, E)$ um grafo direcionado, e (G, W) um grafo ponderado em que $w: E \to \mathbb{Z}^+$.

  Projete uma solução para encontrar o menor caminho monotonicamente crescente entre dois vértices dados.

↳ Caminho: $u_1 \quad u_2 \quad u_3 \ldots u_n$

$w(u_1, u_2) < w(u_2, u_3)$

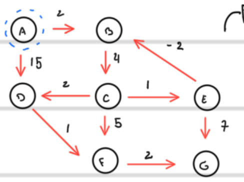
- Seja $G = (V, E)$ um grafo direcionado. Seja (G, W) um grafo ponderado em que $W: E \to \mathbb{Z}^+$.

Projete uma solução para encontrar o menor caminho entre dois vértices s e t em que você tem a opção de "pular" uma aresta específica.

Sua solução deve ser genérica e capaz de funcionar para qualquer aresta escolhida.

28/04/2025

- Encontrar o menor caminho entre A e G.



Se somar +3 e usar Dijkstra → não funciona!

	a	b	c	d	e	f	g
1	0	∞	∞	∞	∞	∞	∞
2	0	2	∞	15	∞	∞	∞
3	0	2	6	15	∞	16	∞
4	0	2	6	8	7	11	18
5	0	2	6	8	7	9	13
6	0	2	6	8	7	9	11

- Ciclo negativo entra em loop
- A quantidade de interações deve ser menor que $|V|$.

05/05/25

Não-Direcionado
- Conexo
  - $\exists$ path (x, y) $\forall x, y \in V$
- Não-Conexo

Grafo-Associado

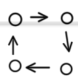
$G_D = (V, E)$ direcionado

$G_A = (V, E')$ será um grafo associado de G se $E' = \{\{u, v\} / (u, v) \in E\}$

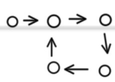
Direcionado
- Conexo
  - Fortemente: $\forall u, v \in V$ $\exists$ path (u, v) e $\exists$ path (v, u)
  - Semi-Fortemente: $\forall u, v \in V$ $\exists$ path (u, v) ou $\exists$ path (v, u)
  - Fracamente
- Não-Conexo

Exemplos:

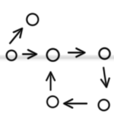
Fortemente:



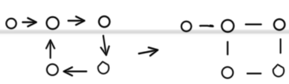
Semi:



Fracamente:



Associado:



'Algoritmos'

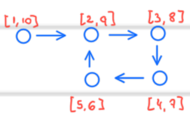
- Fracamente → Se existe um vértice no grafo que o fecho transitivo direto é só o próprio vértice
  ↳ $\Gamma^+(v) = \{v\}$

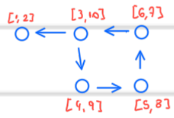
- Fortemente → para cada vértice do grafo o $\Gamma^+(v) = V$ e $\Gamma^-(v) = V$
  ↳ $\forall v \in V \,/\, \Gamma^+(v) = V$ e $\Gamma^-(v) = V$

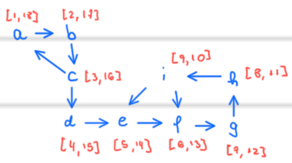
- Semi-Fortemente → $\forall u, v \in V \,/\, v \in \Gamma^+(u)$ ou $u \in \Gamma^+(v)$
  u for 'e' é 'fortemente'

07/05/2025

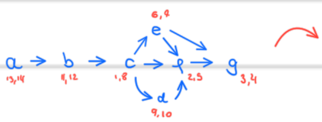
- Projetar algoritmo para achar componentes fortemente conexas.
- Com duas buscas em profundidade







- Como identificar se um grafo é acíclico usando análise de componentes fortemente conexo?



$|CC| = |V|$

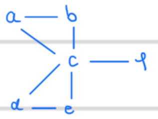
- Seja $G = (V, E)$ um grafo não-direcionado, como encontrar os "componentes fortemente conexos" em G
  ↳ Fazer uma busca

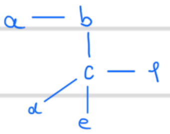
---

08 / 05 / 2025

- Seja $G=(V,E)$ um grafo não-direcionado, e $T \subseteq V$ um subconjunto de vértices. Projete uma solução para identificar uma árvore $H=(V',E')$ em que $H \subseteq G$, $T \subseteq V'$ e $E' \subseteq E$

$G=(V,E)$
$T=\{a,b,c,d,e,f\}$
$T=V=V'$

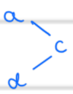




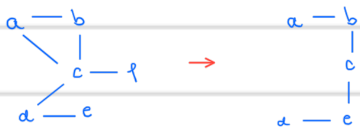
- Árvore de Steiner: É uma árvore que contém necessariamente, alguns vértices de um grafo, é um subgrafo do grafo e contém os vértices terminais mais algum conjunto de vértices.

Fazer um subgrafo apenas com os vértices e arestas necessárias para resolver o problema sem, necessariamente, possuir todos os vértices e arestas do grafo.

ir de 'a' para 'd', minimizando $|E|$:



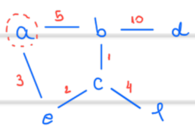
ir de 'a' para 'd', minimizando pesos:



- Branch → Arestas que pertencem a árvore geradora do grafo original.
- Phond → Arestas que não pertencem a árvore geradora.

. Menor Caminho:

- UM vértice para TODOS
- UM vértice para um vértice



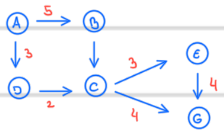
. Qual conjunto de vértices é melhor chegar em 'a' e 'f'?

$a=\{a,e,b,d\}$
$f=\{f,c,b,d\}$
$\rightarrow \forall u \in V$, Dijkstra$(u)$

Fazer em uma única run: (generalização do Dijkstra)

IFT (Image Forest Transform)

12 / 05 / 2025



- Seja A uma raiz e G uma anti-raiz do grupo $G = (V, E)$

Projete uma solução para identificar "quantos" elementos podem ser transportados de A para G.