

# Aula 15 – Principais Problemas tratados com Grafos



DCC405-Estrutura de Dados II  
Prof. Me. Acauan C. Ribeiro

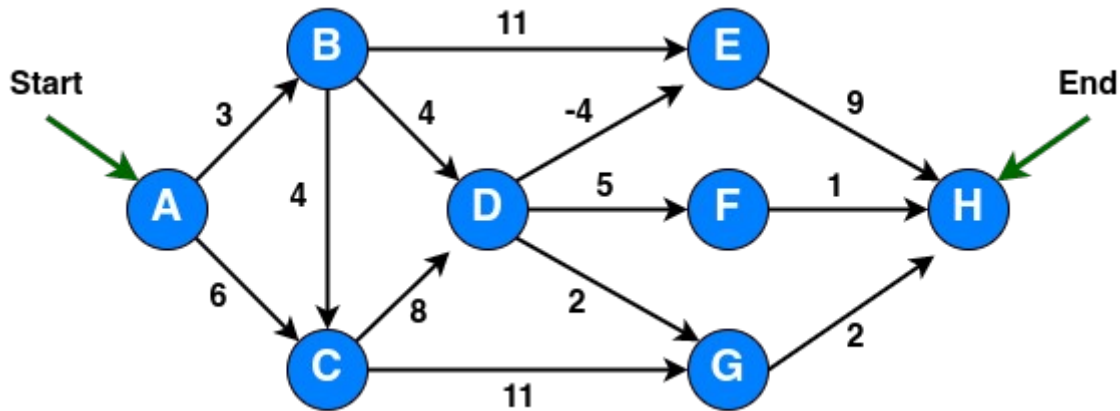
# Problemas Comuns da Teoria dos Grafos

Para os problemas seguintes, faça as seguintes perguntas antes:

- É um grafo direcionado ou não direcionado?
- As arestas do grafo possui pesos?
- No grafo que vai ser construído as arestas vão ser esparsas ou densas?
- Devo utilizar uma matriz de adjacência, lista de adjacência, lista de arestas ou outra estrutura para representar o grafo de maneira eficiente?

# Problema do Menor Caminho (Shortest path problem)

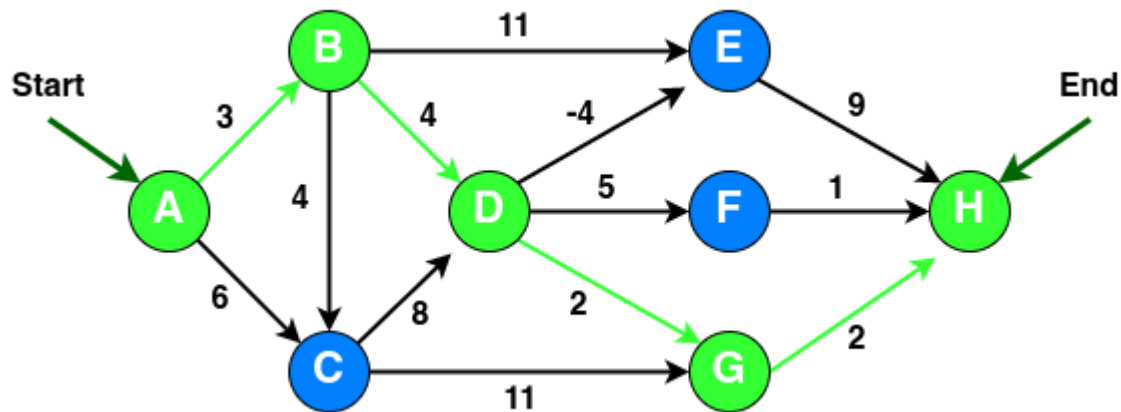
Dado um grafo ponderado, encontre o menor caminho de arestas entre o vértice A e o vértice B.



**Algoritmos:** BFS (unweighted graph), Dijkstra's, Bellman Ford, Floyd-Warshall, A\* e vários outros.

# Problema do Menor Caminho (Shortest path problem)

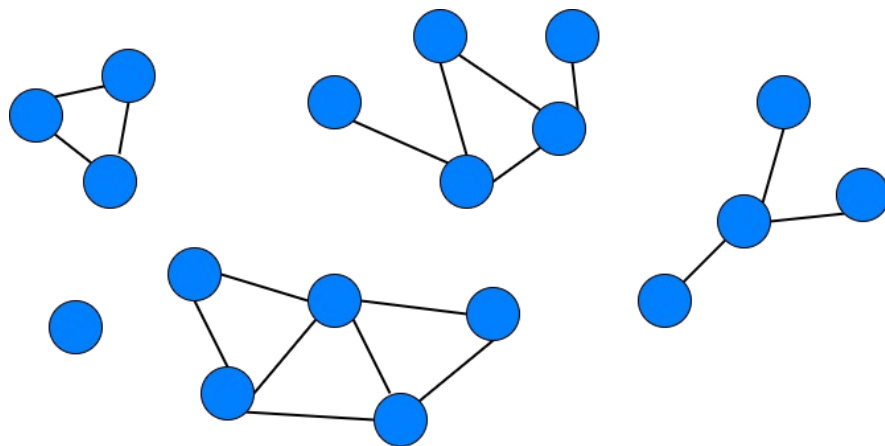
Dado um grafo ponderado, encontre o menor caminho de arestas entre o vértice A e o vértice B.



**Algoritmos:** BFS (unweighted graph), Dijkstra's, Bellman Ford, Floyd-Warshall, A\* e vários outros.

# Problema: Conectividade

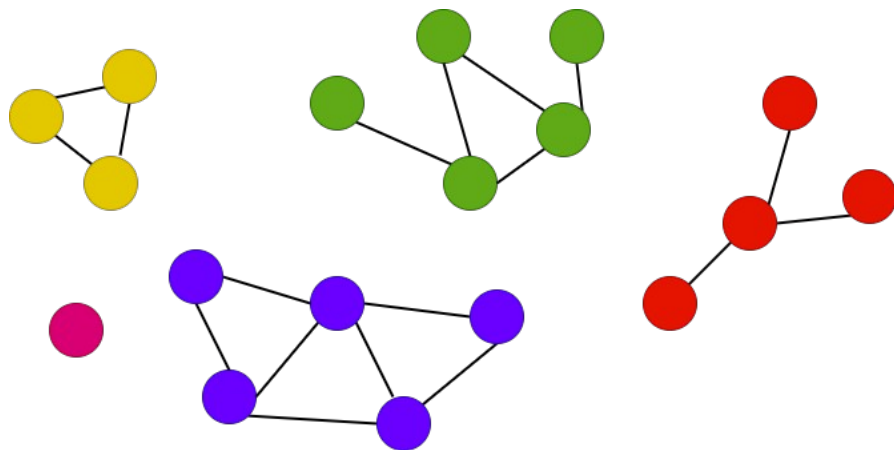
- Existe algum caminho entre os vértices A e B?



**Solução típica:** Utilize a **busca pela união (Union Find)** das estruturas de dados ou qualquer algoritmo de busca (Exemplo.: DFS)

# Problema: Conectividade

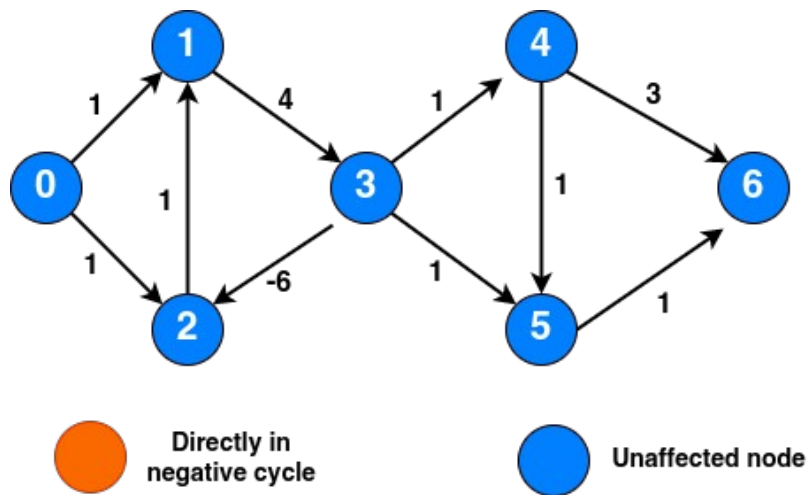
- Existe algum caminho entre os vértices A e B?



**Solução típica:** Utilize a **busca pela união (Union Find)** das estruturas de dados ou qualquer algoritmo de busca (Exemplo.: DFS)

# Problema: Detectar Ciclos Negativos

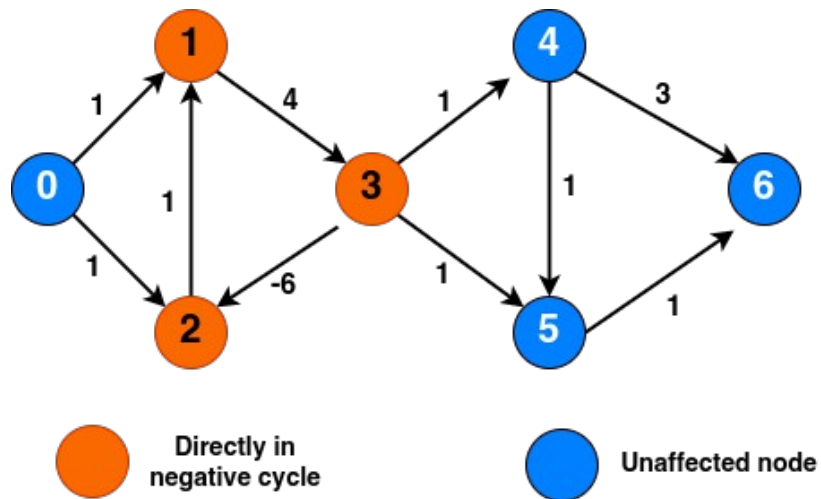
O meu dígrafo com peso nas arestas possui algum ciclo negativo? Se sim, onde?



**Algoritmos:** Bellman Ford e Floyd-Warshall

# Problema: Detectar Ciclos Negativos

O meu dígrafo com peso nas arestas possui algum ciclo negativo? Se sim, onde?

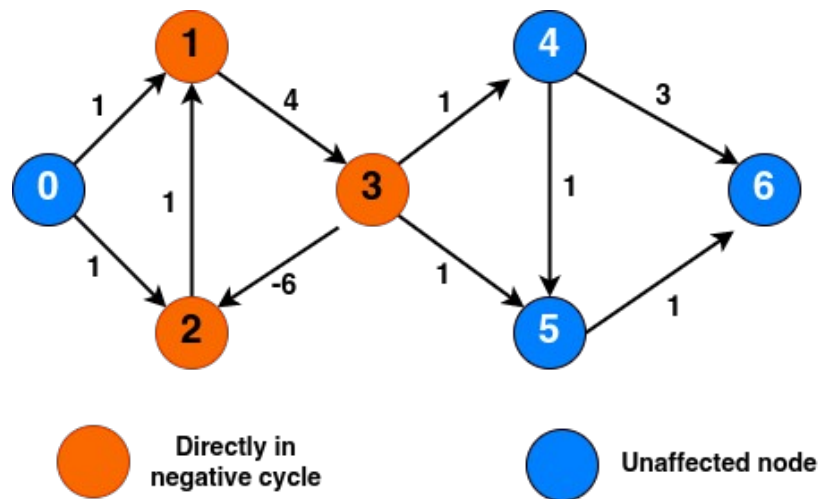


**Algoritmos:** Bellman Ford e Floyd-Warshall



# Problema: Detectar Ciclos Negativos

O meu dígrafo com peso nas arestas possui algum ciclo negativo? Se sim, onde?

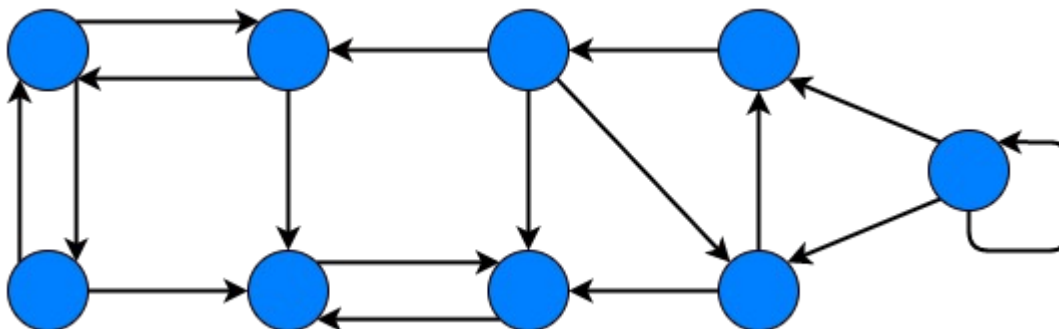


**Algoritmos:** Bellman Ford e Floyd-Warshall

Um cenário possível de aplicação de identificação de ciclos negativos pode ser observado na compra e venda de moedas na bolsa de valores. Um ação chamada “**arbitragem**” faz a simulação de compra e venda (troca) de moedas tendo em consideração o valor do câmbio entre mercados distintos, ao final você pode ter ciclos ou caminhos que lhe apresenta perda, lucro ou margem igualitária.

# Problema: Componentes Fortemente Conectados

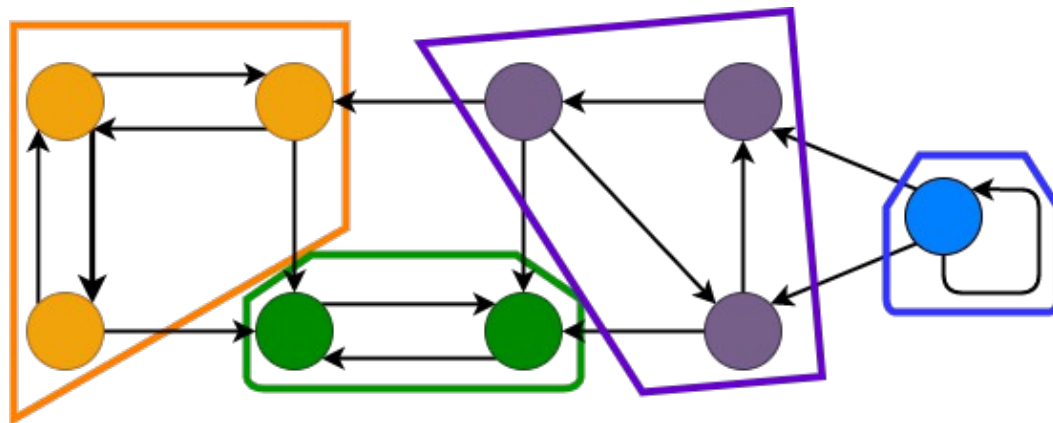
Componentes Fortemente Conectados (**Strongly Connected Components – SCCs**) podem ser pensados como **self-contained cycles** em um grafo direcionado onde cada vértice em um ciclo pode ser alcançado por qualquer outro vértice no mesmo ciclo.



**Algoritmos:** Tarjan's e Kosaraju's algorithm.

# Problema: Componentes Fortemente Conectados

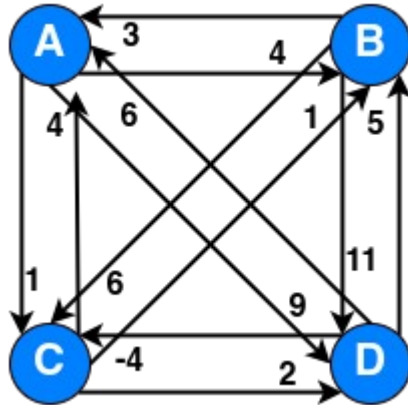
Componentes Fortemente Conectados (**Strongly Connected Components – SCCs**) podem ser pensados como **self-contained cycles** em um grafo direcionado onde cada vértice em um ciclo pode ser alcançado por qualquer outro vértice no mesmo ciclo.



**Algoritmos:** Tarjan's e Kosaraju's algorithm.

# O problema do Caixeiro Viajante

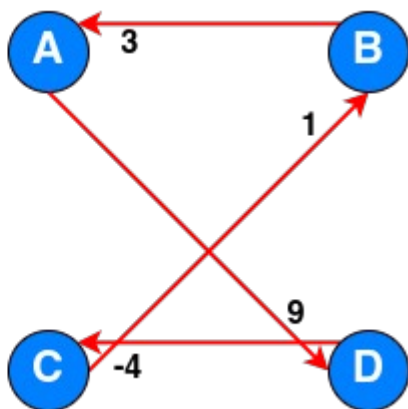
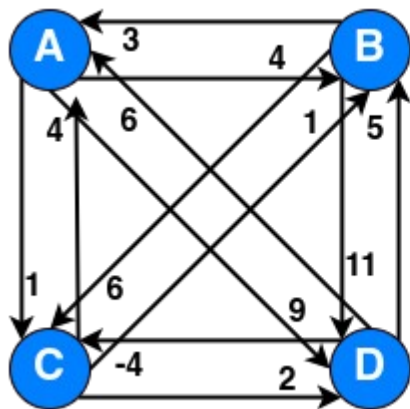
“Dado uma lista de cidades e as distâncias entre cada par de cidades, qual é o menor rota possível que visite cada cidade exatamente uma vez e retorne a cidade de origem” – Wiki.



**Algoritmos:** Held-Karp, Ramificar e limitar (Branch and bound) e vários algoritmos de aproximação

# O Problema do Caixeiro Viajante (TSP – Traveling Salesman Problem)

“Dado uma lista de cidades e as distâncias entre cada par de cidades, qual é o menor rota possível que visite cada cidade exatamente uma vez e retorne a cidade de origem” – Wiki.

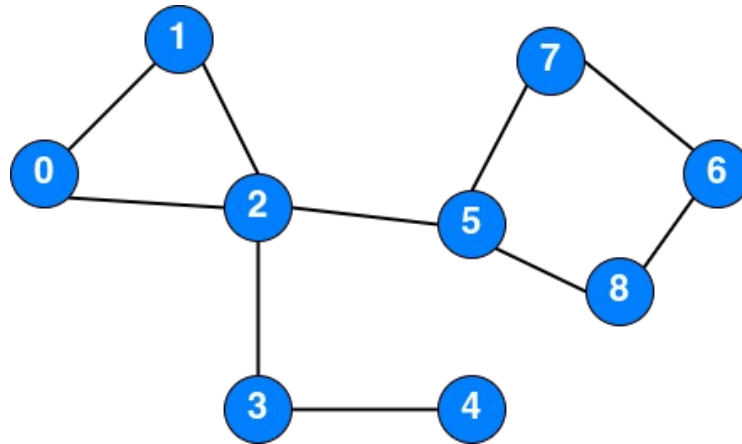


Esse é um problema **NP-difícil**. O que significa que é computacionalmente desafiador.

**Algoritmos:** Held-Karp, Ramificar e limitar (Branch and bound) e várias outras técnicas de Programação Dinâmica e Algoritmos de Aproximação e Otimização (Colônia de Formigas)

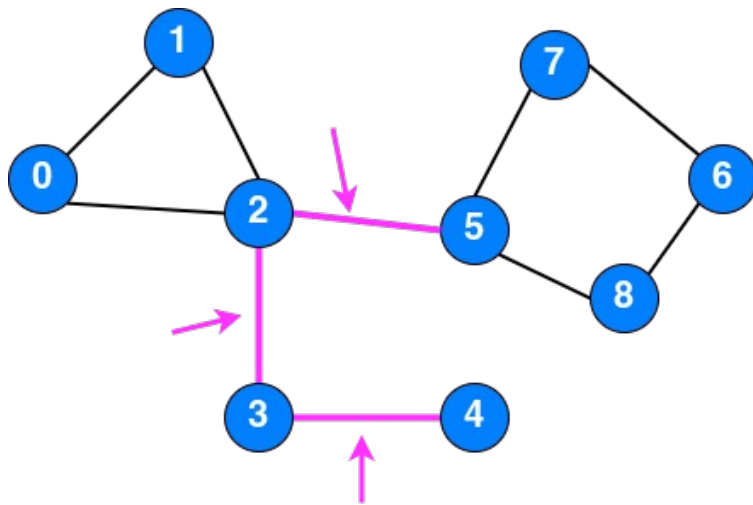
# Problema: Pontes / Bridges

Uma **ponte** / **cut edge** é qualquer aresta em um grafo no qual a sua remoção aumenta o número de componentes conectados.



# Problema: Pontes / Bridges

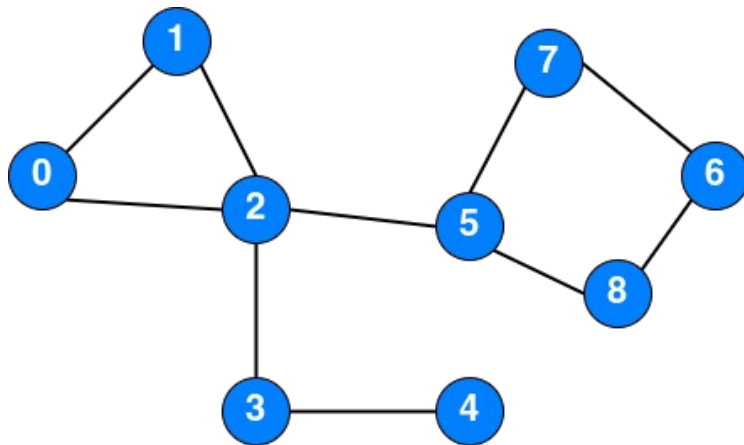
Uma **ponte** / **cut edge** é qualquer aresta em um grafo no qual a sua remoção aumenta o número de **componentes conectados**.



A **Pontes** são importantes na Teoria dos Grafos, devido as mesmas oferecerem uma aviso (indicação) para “**pontos fracos**”, “**gargalos**” ou “**vulnerabilidades**” em um grafo.

# Problema: Pontos de Articulação

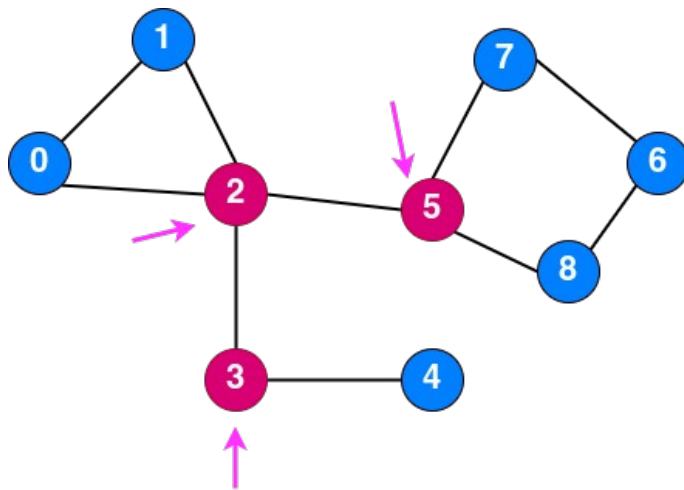
Um **Ponto de Articulação** / **cut vertex** é qualquer nó no grafo onde a remoção aumenta o número de **componentes conectados**.





# Problema: Pontos de Articulação

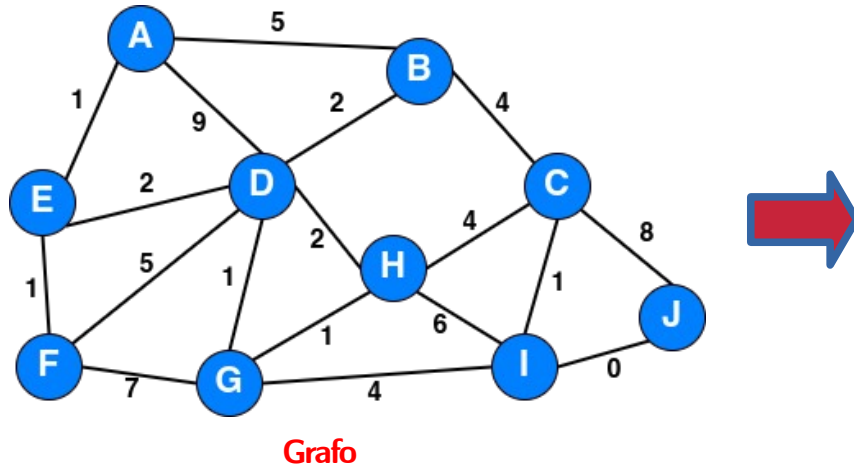
Um **Ponto de Articulação** / **cut vertex** é qualquer nó no grafo onde a remoção aumenta o número de **componentes conectados**.



Os pontos de articulação são importantes na teoria dos grafos devido aos mesmos apontarem para “pontos fracos”, “gargalos” ou “vulnerabilidades” em um grafo.

# Árvore Geradora Mínima (Minimum Spanning Tree - MST)

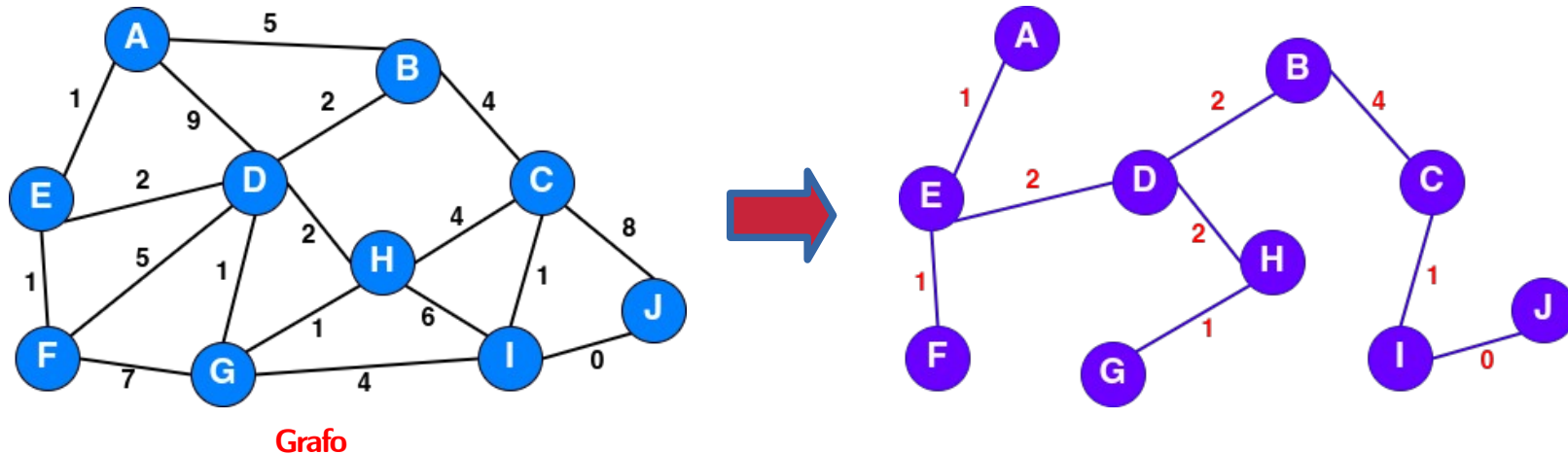
Uma Árvore Geradora Mínima (MST) é um subconjunto de arestas de um grafo conectado, ponderado que faz com que esse grafo seja totalmente conectado, sem gerar nenhum ciclo e com um custo mínimo quando somado todas essas arestas que fazem esta conexão.



**Algoritmos:** Kruskal's, Prim's & Boruvka's algorithm.

# Árvore Geradora Mínima (Minimum Spanning Tree - MST)

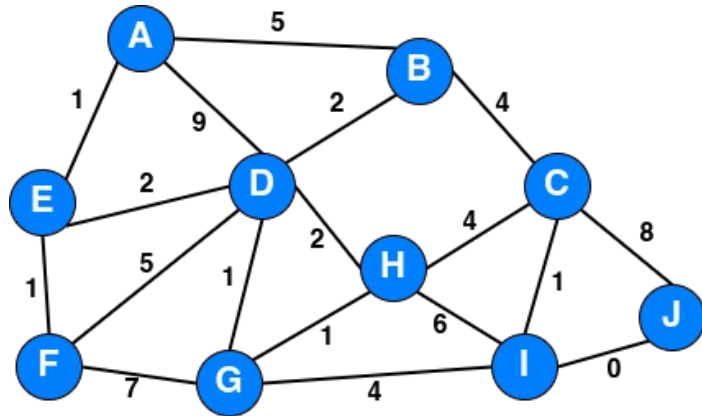
Uma Árvore Geradora Mínima (MST) é um subconjunto de arestas de um grafo conectado, ponderado que faz com que esse grafo seja totalmente conectado, sem gerar nenhum ciclo e com um custo mínimo quando somado todas essas arestas que fazem esta conexão.



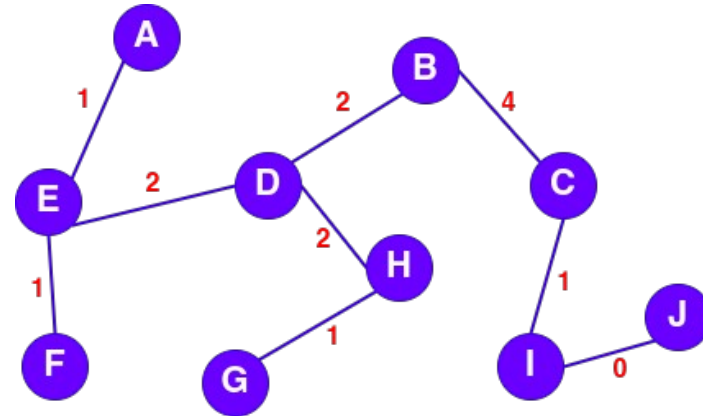
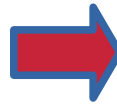
**Algoritmos:** Kruskal's, Prim's & Boruvka's algorithm.

# Árvore Geradora Mínima (Minimum Spanning Tree - MST)

Uma Árvore Geradora Mínima (MST) é um subconjunto de arestas de um grafo conectado, ponderado que faz com que esse grafo seja totalmente conectado, sem gerar nenhum ciclo e com um custo mínimo quando somado todas essas arestas que fazem esta conexão.



Grafo

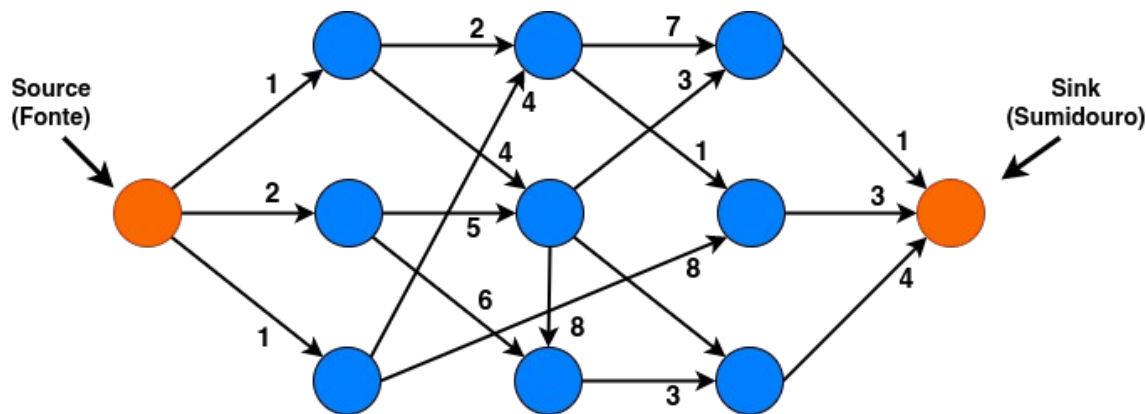


Essa MST tem um **peso total de 14**.  
Note que existem outras MSTs possíveis no mesmo grafo.

**Algoritmos:** Kruskal's, Prim's & Boruvka's algorithm.

# Problema: Fluxo Máximo em Rede (Network flow: Max flow)

Q: Com uma fonte de entrada infinita, quanto "fluxo" podemos enviar pela rede?



Suponha que as arestas são rodovias com carros, tubulações com água ou corredores lotados de pessoas. O Fluxo representa o volume de água suportado para passar dentro dos canos, número de carros uma rodovia suporta trafegar e o volume máximo de pessoas que passam por um corredor

**Algoritmos:** Ford-Fulkerson, Edmons-Karp & Dinic's algorithm.

# Top 5 Most Common Graph Algorithms for Coding Interviews

Assistir: <https://youtu.be/utDu3Q7Flrw>

# Referências

- William Fiset - Software Engineer at Google

<https://www.linkedin.com/in/williamfiset/>