Grafos

Caminhos mínimos

Prof. Edson Alves

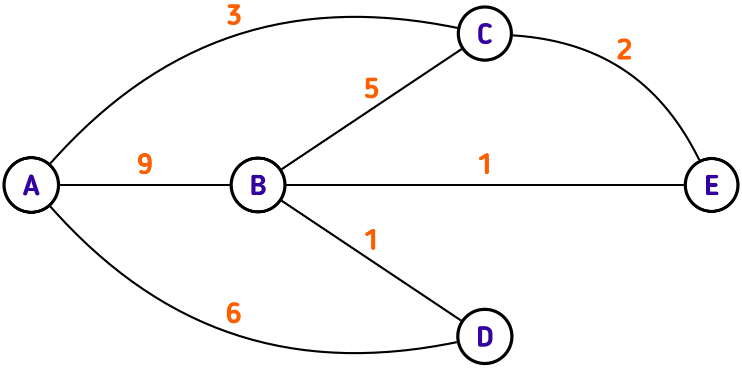
Faculdade UnB Gama

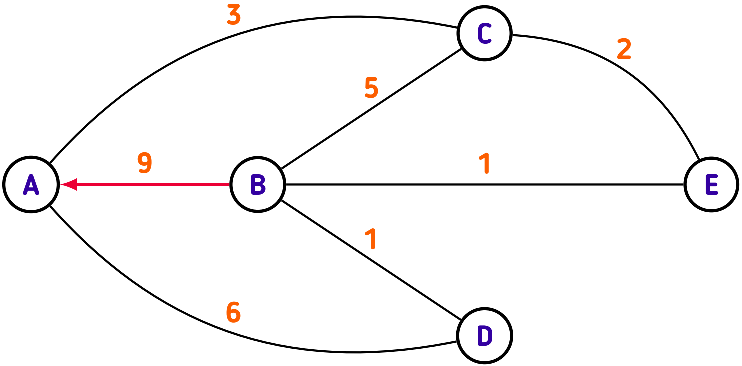
Caminhos mínimos

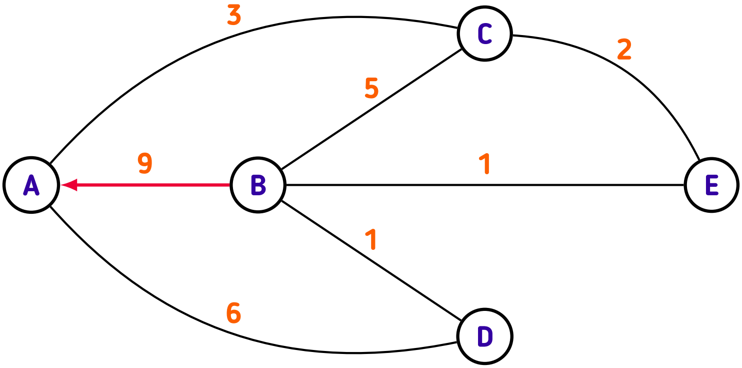
Seja p um caminho entre os vértices u e v do grafo G. Dizemos que p é um **caminho mínimo** de u a v se, para qualquer caminho q de u a v, vale que

$$\sum_{e_i \in p} w(e_i) \leq \sum_{e_j \in q} w(e_j)$$

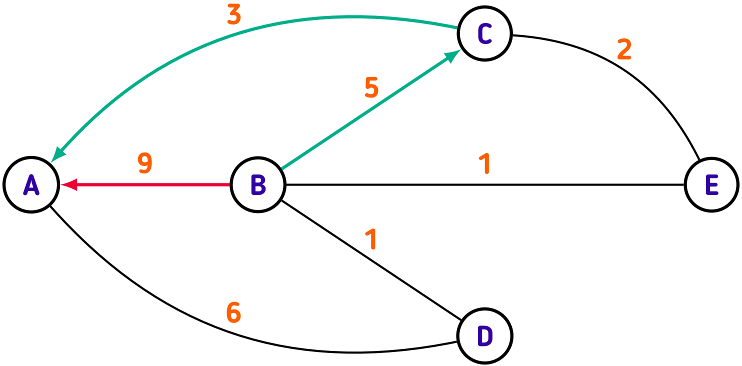
onde $w(e)$ é o peso da aresta e.


3
C
2
5
A
9
B
1
E
1
6
D

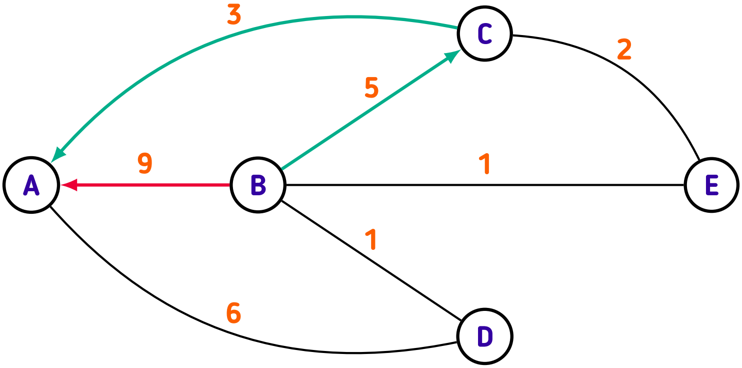

A
B
C
D
E
3
2
5
9
1
1
6



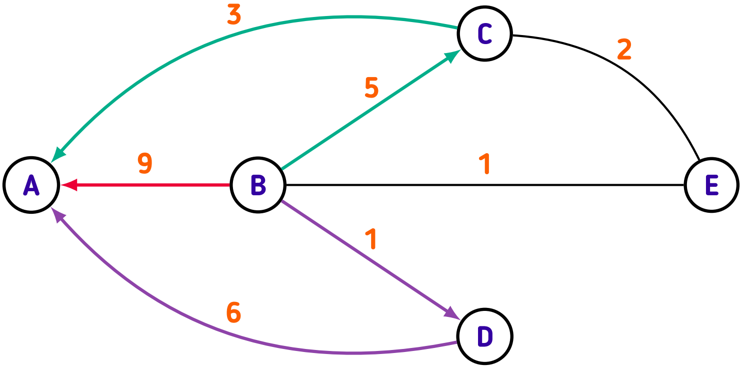
$$\sum_{e \in p_1} w(e) = 9$$



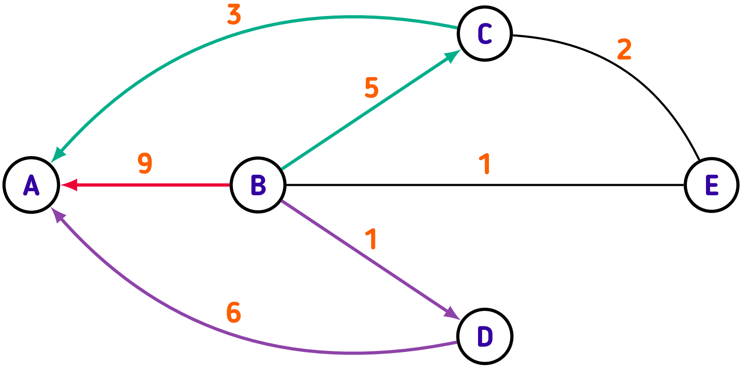
$$\sum_{e \in p_1} w(e) = 9$$



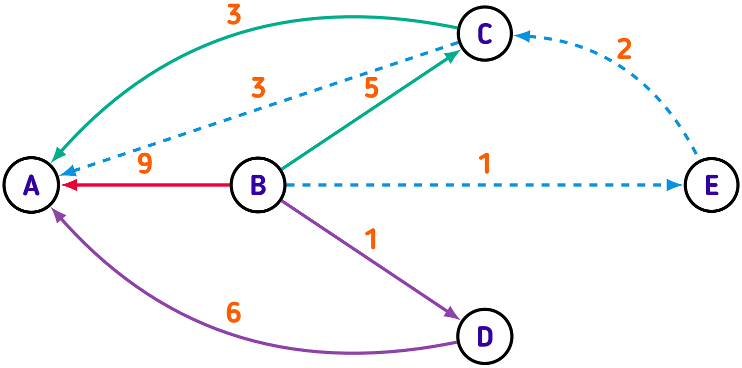
$$\sum_{e \in p_1} w(e) = 9 \qquad \sum_{e \in p_2} w(e) = 8$$



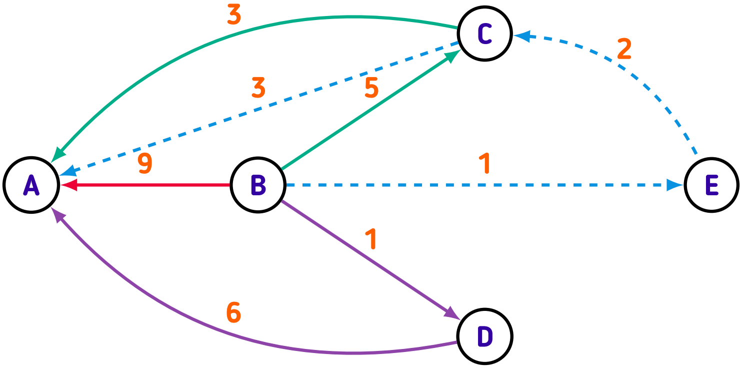
$$\sum_{e \in p_1} w(e) = 9 \qquad \sum_{e \in p_2} w(e) = 8$$



$$\sum_{e \in p_1} w(e) = 9 \qquad \sum_{e \in p_2} w(e) = 8 \qquad \sum_{e \in p_3} w(e) = 7$$



$$\sum_{e\in p_1} w(e) = 9 \qquad \sum_{e\in p_2} w(e) = 8 \qquad \sum_{e\in p_3} w(e) = 7$$



$$\sum_{e\in p_1} w(e) = 9 \qquad \sum_{e\in p_2} w(e) = 8 \qquad \sum_{e\in p_3} w(e) = 7 \qquad \sum_{e\in p_3} w(e) = 6$$

Caminhos mínimos em grafos não-ponderado

Caminhos mínimos em grafos não-ponderado

- Se G é não-ponderado, o custo de um caminho pode ser medido em arestas

Caminhos mínimos em grafos não-ponderado

- $\star$ Se G é não-ponderado, o custo de um caminho pode ser medido em arestas
- $\star$ Isto equivale a considerar que todas as arestas de G tem peso 1

Caminhos mínimos em grafos não-ponderado

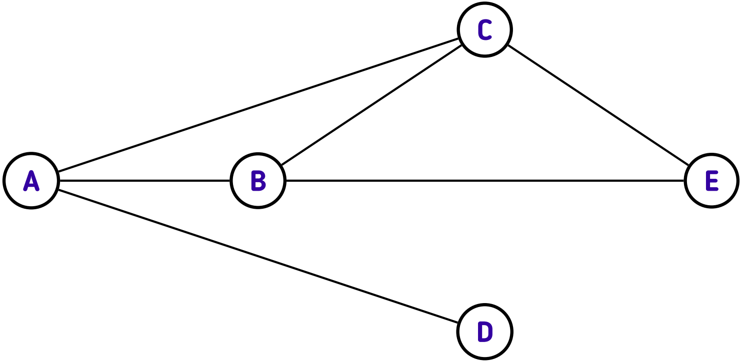
$\star$ Se G é não-ponderado, o custo de um caminho pode ser medido em arestas

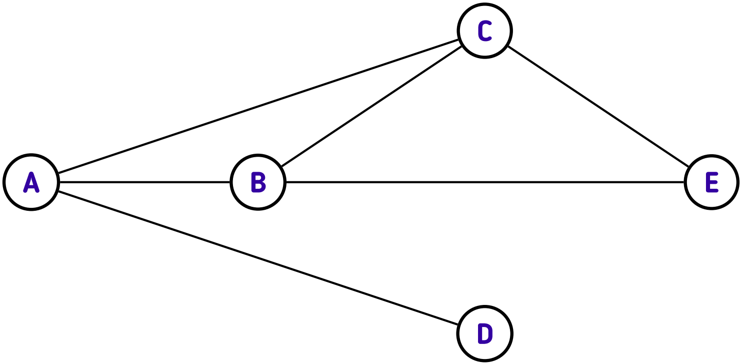
$\star$ Isto equivale a considerar que todas as arestas de G tem peso 1

$\star$ Neste caso, uma BFS pode determinar a distância mínima entre o vértice de partida s e todos os vértices de G

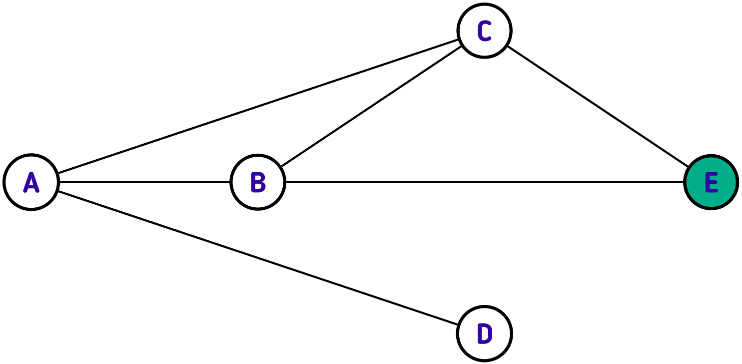
Caminhos mínimos em grafos não-ponderado

- $\star$ Se G é não-ponderado, o custo de um caminho pode ser medido em arestas
- $\star$ Isto equivale a considerar que todas as arestas de G tem peso 1
- $\star$ Neste caso, uma BFS pode determinar a distância mínima entre o vértice de partida s e todos os vértices de G
- $\star$ A BFS também poder ser usada em grafos cujas arestas tem todas o mesmo peso c

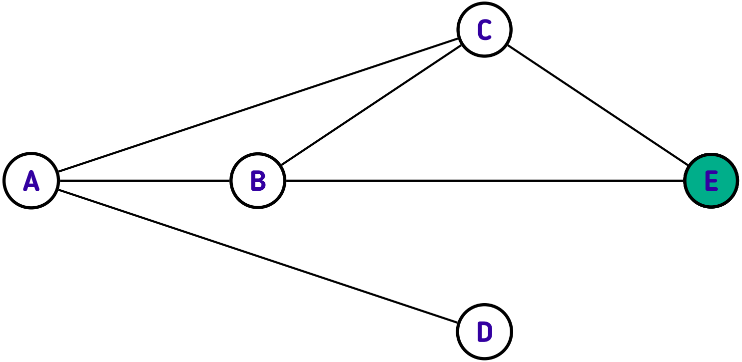

A
B
C
D
E



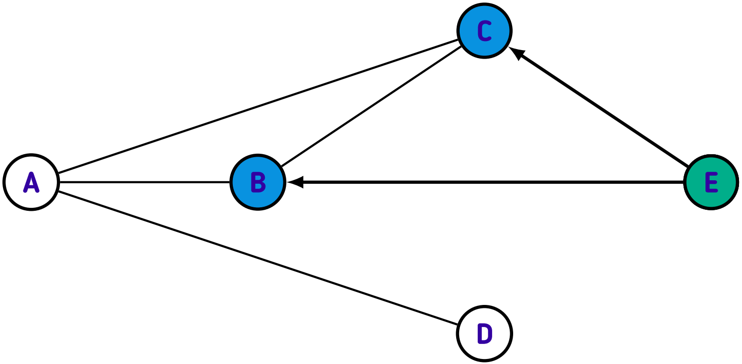
	A	B	C	D	E
dist(u, **E**)	∞	∞	∞	∞	∞



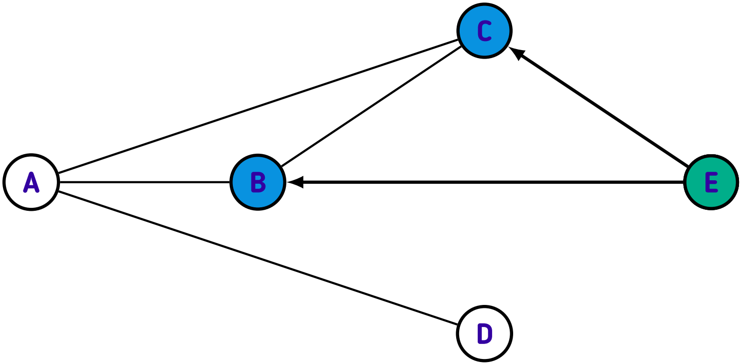
	A	B	C	D	E
$\mathsf{dist}(u, \mathbf{E})$	∞	∞	∞	∞	∞



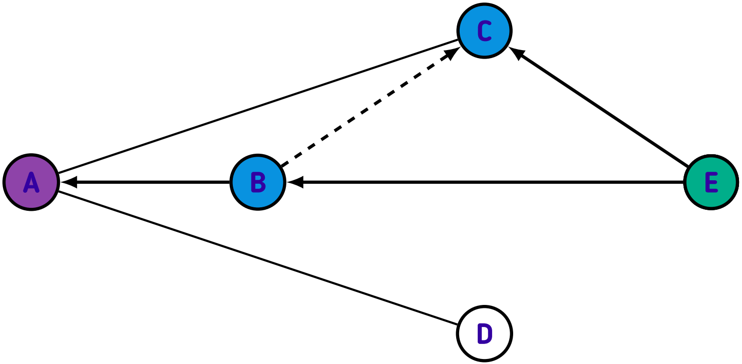
	A	B	C	D	E
dist$(u,$ **E**$)$	∞	∞	∞	∞	0



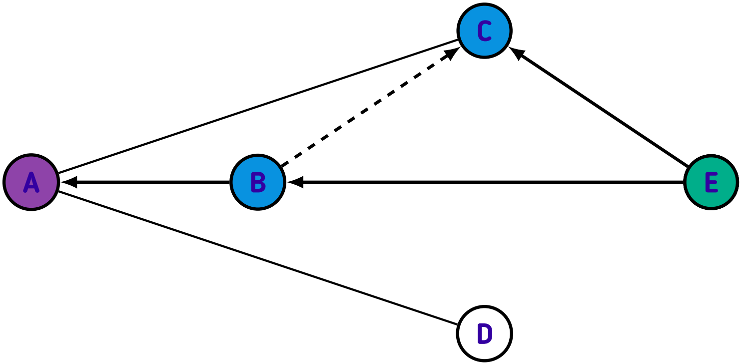
	A	B	C	D	E
$\mathrm{dist}(u, \mathbf{E})$	∞	∞	∞	∞	0



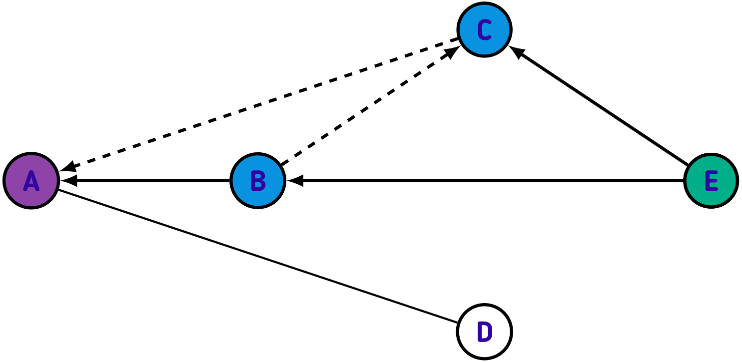
	A	B	C	D	E
dist$(u, \mathbf{E})$	∞	1	1	∞	0



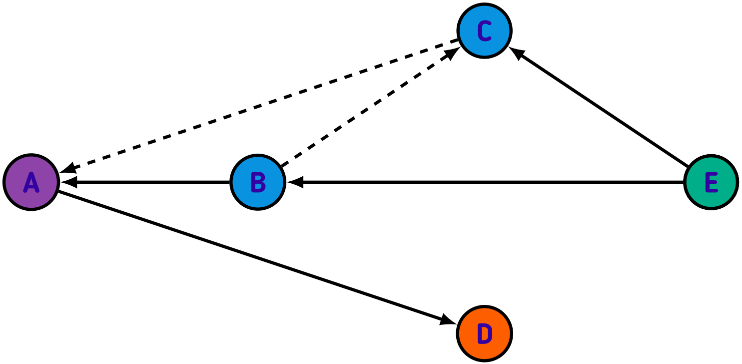
	A	B	C	D	E
dist$(u,$ **E**$)$	∞	1	1	∞	0



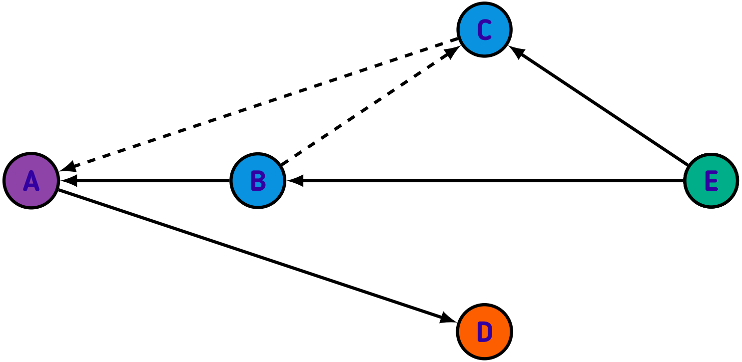
	A	B	C	D	E
dist$(u, \mathbf{E})$	2	1	1	∞	0



	A	B	C	D	E
dist$(u, \textbf{E})$	2	1	1	∞	0



	A	B	C	D	E
$\mathsf{dist}(u, \textbf{E})$	2	1	1	∞	0



	A	B	C	D	E
dist$(u, \mathbf{E})$	2	1	1	3	0

```cpp
vector<int> min_dist(int s, int N, int c = 1) {
    vector<int> dist(N + 1, oo);
    queue<int> q;

    dist[s] = 0; q.push(s);

    while (not q.empty())
    {
        auto u = q.front(); q.pop();

        for (auto v : adj[u]) {
            if (dist[v] == oo) {
                dist[v] = dist[u] + c;
                q.push(v);
            }
        }
    }

    return dist;
}
```

Caminhos mínimos em grafos ponderados

Caminhos mínimos em grafos ponderados

$\star$ **Se G é ponderado, existem outros algoritmos que permitem determinar as distâncias mínimas de um vértice de partida s aos demais vértices de G**

Caminhos mínimos em grafos ponderados

⋆ Se G é ponderado, existem outros algoritmos que permitem determinar as distâncias mínimas de um vértice de partida s aos demais vértices de G

⋆ No caso geral, deve ser empregado o algoritmo de Bellman-Ford

Caminhos mínimos em grafos ponderados

$\star$ **Se G é ponderado, existem outros algoritmos que permitem determinar as distâncias mínimas de um vértice de partida s aos demais vértices de G**

$\star$ **No caso geral, deve ser empregado o algoritmo de Bellman-Ford**

$\star$ **Se não há arestas com pesos negativos, pode-se usar o algoritmo de Dijkstra**

Caminhos mínimos em grafos ponderados

$\star$ Se G é ponderado, existem outros algoritmos que permitem determinar as distâncias mínimas de um vértice de partida s aos demais vértices de G

$\star$ No caso geral, deve ser empregado o algoritmo de Bellman-Ford

$\star$ Se não há arestas com pesos negativos, pode-se usar o algoritmo de Dijkstra

$\star$ Se os pesos das arestas são ou 0 ou 1, a BFS 0/1 é uma alternativa eficiente

Caminhos mínimos em grafos ponderados

$\star$ Se G é ponderado, existem outros algoritmos que permitem determinar as distâncias mínimas de um vértice de partida s aos demais vértices de G

$\star$ No caso geral, deve ser empregado o algoritmo de Bellman-Ford

$\star$ Se não há arestas com pesos negativos, pode-se usar o algoritmo de Dijkstra

$\star$ Se os pesos das arestas são ou 0 ou 1, a BFS 0/1 é uma alternativa eficiente

$\star$ Para determinar todas as distâncias mínimas entre todos os pares de vértices de G, há o algoritmo de Floyd-Warshall

Problemas sugeridos

1. AtCoder Beginner Contest 088 – Problem D: Repainting
2. Codeforces Beta Round #3 – Problem A: Shortest path of the king
3. OJ 10000 – Longest Paths
4. OJ 10959 – The Party, Part I

Referências

1. **HALIM**, **Felix**; **HALIM**, **Steve**. *Competitive Programming 3*, **2010.**
2. **LAAKSONEN**, **Antti**. *Competitive Programmer's Handbook*, **2018.**
3. **SKIENA**, **Steven**; **REVILLA**, **Miguel**. *Programming Challenges*, **2003.**