

Grafos

Algoritmo de Bellman-Ford

Prof. Edson Alves

Faculdade UnB Gama

Proponentes

Proponentes



Lester Randolph Ford Jr.
(1956)

Proponentes



Lester Randolph Ford Jr.
(1956)



Richard Ernest Bellman
(1958)

Características do algoritmo de Bellman-Ford

Características do algoritmo de Bellman-Ford

- ★ Computa o caminho mínimo de todos os vértices de $G(V, E)$ a um dado nó s

Características do algoritmo de Bellman-Ford

- ★ Computa o caminho mínimo de todos os vértices de $G(V, E)$ a um dado nó s
- ★ É capaz de processar arestas negativas

Características do algoritmo de Bellman-Ford

- ★ Computa o caminho mínimo de todos os vértices de $G(V, E)$ a um dado nó s
- ★ É capaz de processar arestas negativas
- ★ Não processa, mas identifica ciclos negativos

Características do algoritmo de Bellman-Ford

- ★ Computa o caminho mínimo de todos os vértices de $G(V, E)$ a um dado nó s
- ★ É capaz de processar arestas negativas
- ★ Não processa, mas identifica ciclos negativos
- ★ Complexidade: $O(VE)$

Pseudocódigo

Pseudocódigo

Entrada: um grafo $G(V, E)$ e um vértice $s \in V$

Saída: um vetor d tal que $d[u]$ é a distância mínima em G entre s e u

Pseudocódigo

Entrada: um grafo $G(V, E)$ e um vértice $s \in V$

Saída: um vetor d tal que $d[u]$ é a distância mínima em G entre s e u

1. Faça $d[s] = 0$ e $d[u] = \infty$ para todos vértices $u \in V$ tais que $u \neq s$

Pseudocódigo

Entrada: um grafo $G(V, E)$ e um vértice $s \in V$

Saída: um vetor d tal que $d[u]$ é a distância mínima em G entre s e u

1. **Faça** $d[s] = 0$ e $d[u] = \infty$ para todos vértices $u \in V$ tais que $u \neq s$
2. **Para cada aresta** $(u, v, w) \in E$, **se** $d[u] + w < d[v]$, **faça** $d[v] = d[u] + w$

Pseudocódigo

Entrada: um grafo $G(V, E)$ e um vértice $s \in V$

Saída: um vetor d tal que $d[u]$ é a distância mínima em G entre s e u

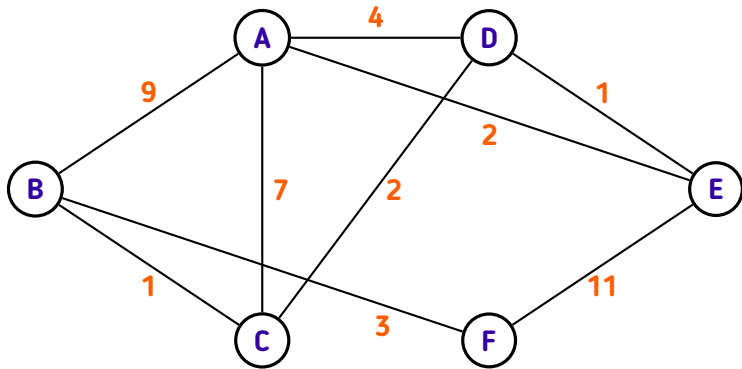
1. **Faça** $d[s] = 0$ e $d[u] = \infty$ para todos vértices $u \in V$ tais que $u \neq s$
2. **Para cada aresta** $(u, v, w) \in E$, **se** $d[u] + w < d[v]$, **faça** $d[v] = d[u] + w$
3. **Se o vetor** d **foi atualizado ao menos uma vez, volte ao passo 2.**

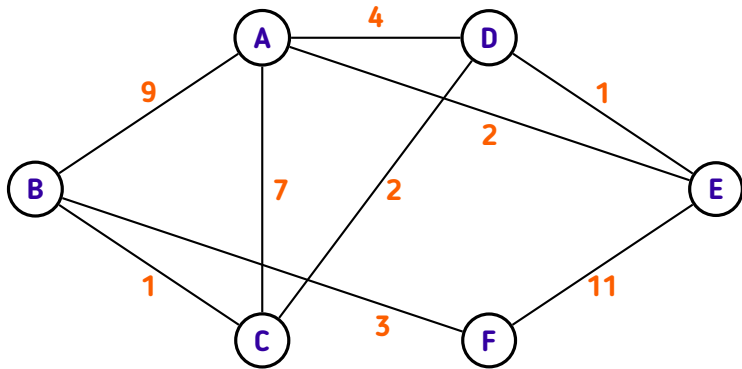
Pseudocódigo

Entrada: um grafo $G(V, E)$ e um vértice $s \in V$

Saída: um vetor d tal que $d[u]$ é a distância mínima em G entre s e u

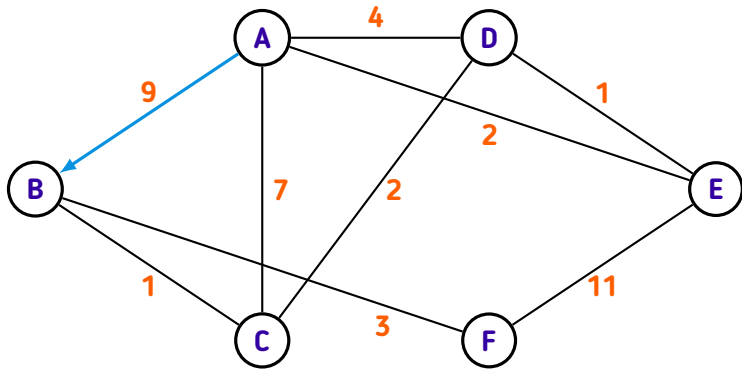
1. Faça $d[s] = 0$ e $d[u] = \infty$ para todos vértices $u \in V$ tais que $u \neq s$
2. Para cada aresta $(u, v, w) \in E$, se $d[u] + w < d[v]$, faça $d[v] = d[u] + w$
3. Se o vetor d foi atualizado ao menos uma vez, volte ao passo 2.
4. Retorne d



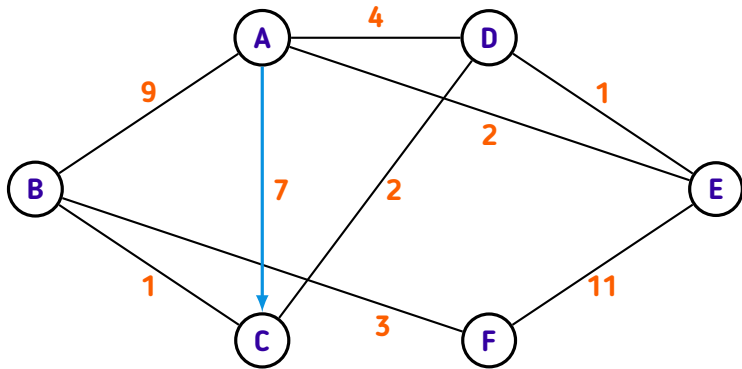


$\text{dist}(u, \mathbf{A})$

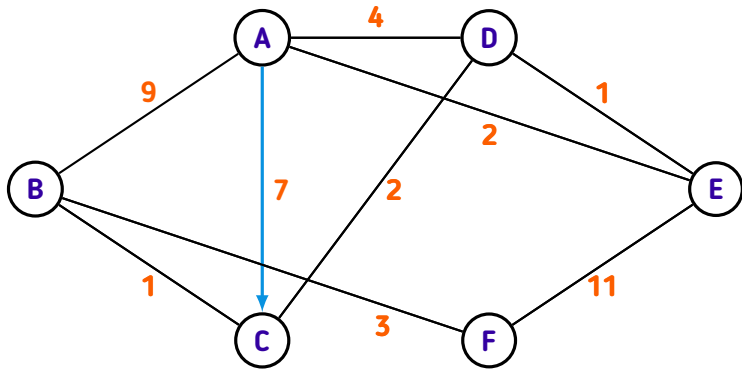
A	B	C	D	E	F
0	∞	∞	∞	∞	∞



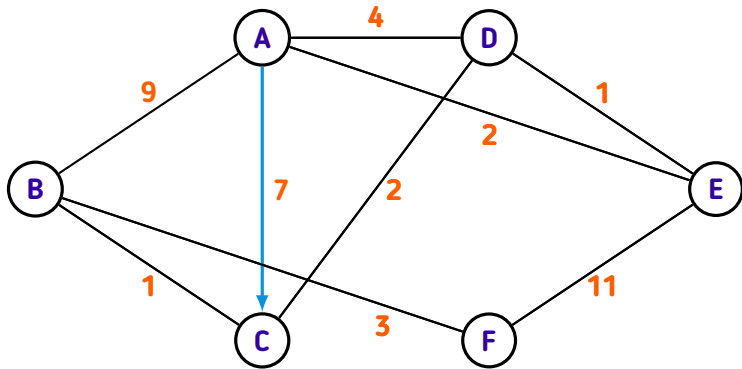
	A	B	C	D	E	F
$\text{dist}(u, \mathbf{A})$	0	9	∞	∞	∞	∞



	A	B	C	D	E	F
$\text{dist}(u, \mathbf{A})$	0	9	7	∞	∞	∞



	A	B	C	D	E	F
$\text{dist}(u, \mathbf{A})$	0	9	7	∞	∞	∞



	A	B	C	D	E	F
$\text{dist}(u, \mathbf{A})$	0	9	7	∞	∞	∞

```
vector<int> bellman_ford(int s, int N, const vector<edge>& edges)
{
    const int oo { 1000000010 };

    vector<int> dist(N + 1, oo);
    dist[s] = 0;

    for (int i = 1; i <= N - 1; i++)
        for (auto [u, v, w] : edges)
            dist[v] = min(dist[v], dist[u] + w);

    return dist;
}
```

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.
4. Wikipédia, *Bellman-Ford algorithm*. Acesso em 07/07/2021.
5. Wikipédia, *L. R. Ford Jr.* Acesso em 07/07/2021.
6. Wikipédia, *Richard E. Bellman*. Acesso em 07/07/2021.