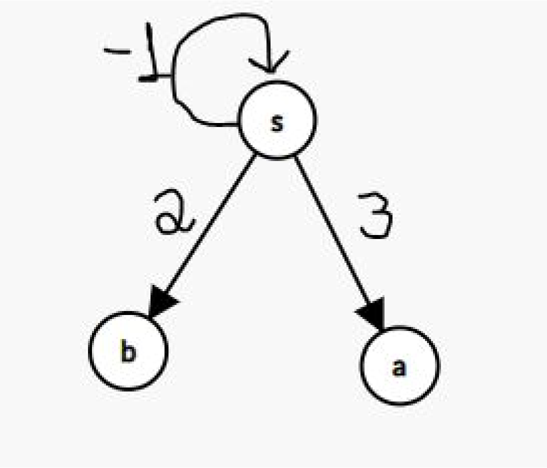
[DESAFIO] Módulo IV -- 414624

DESAFIO 01

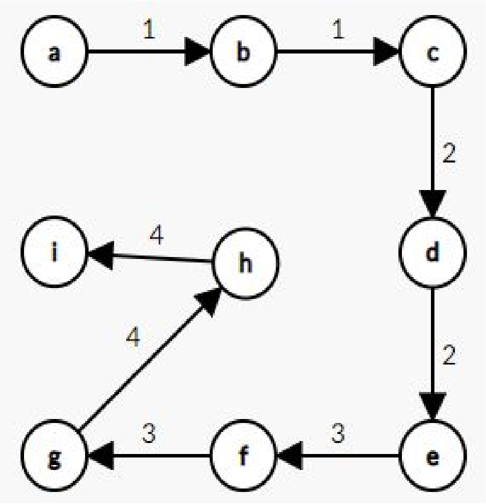


Vértice fonte: **s**

A distância de s para os outros vértices são:
d[b] = 2
d[a] = 3

DESAFIO 02
Nos casos onde todas as arestas têm pesos iguais e positivos.
Como o ABL percorre o menor número de arestas para visitar todos os vértices e o atributo d[v] só é modificado uma única vez. Portanto, para ser um caminho mínimo, os pesos das arestas devem ser iguais.

DESAFIO 03



Vértice fonte: a
Ordem:
{ (a, b), (b, c), (c, d), (d, e), (e, f), (f, g), (g, h), (h, i) }

Início:
(a, d[0]), (b, d[∞]), (c, d[∞]), (d, d[∞]), (e, d[∞]), (f, d[∞]), (g, d[∞]), (h, d[∞]), (i, d[∞])

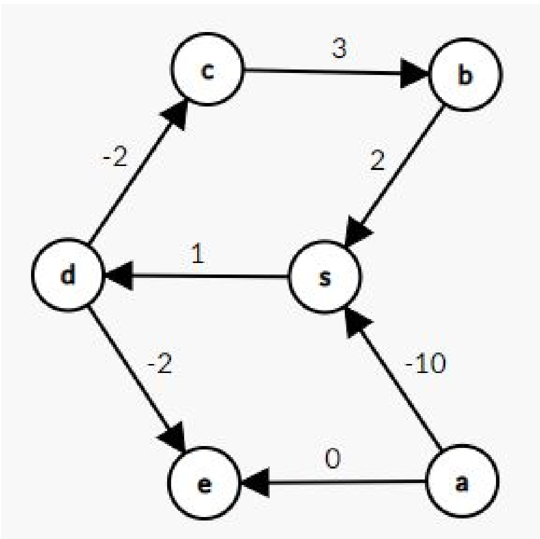
Primeira iteração:
(a, d[0]), (b, d[1]), (c, d[2]), (d, d[4]), (e, d[6]), (f, d[9]), (g, d[12]), (h, d[16]), (i, d[20])

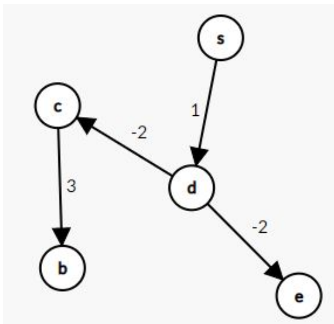
DESAFIO 04

O algoritmo Bellman-Ford não calcula corretamente o caminho mínimo entre os vértices em grafos que possuem ciclos negativos.
Suponha um caminho p que possui um ciclo negativo e com peso **w(p).**
Suponha que o ciclo negativo seja repetido outra vez e que o peso desse ciclo negativo seja **w(c)**.
Temos que **w(p') = w(p) + w(c)**, então **w(p) < w(p')**.
Repetindo novamente essa ideia, podemos obter um valor menor.
Como a definição de caminho mínimo é o caminho de menor peso, por essa razão, não faz sentido examinar a distância de caminhos mínimos entre vértices u e v quando u alcança algum ciclo negativo no grafo.

DESAFIO 05
Se o valor de k for menor que n, podemos alterar no loop for para que ele seja executado k vezes.
Assim, como k < n, temos que a complexidade **O(k*m) < O(n*m).**

DESAFIO 06

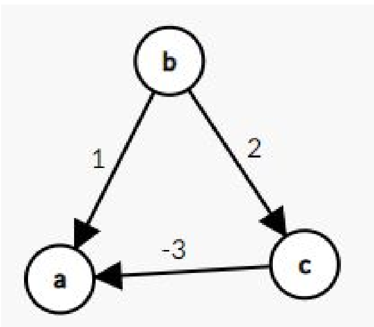




← Gpi

Vértice fonte: s
S = { s, d, c, e, b, a }

DESAFIO 07



Vértice fonte: b
S = { b, a, c }
Pelo conceito de caminho mínimo, d[a] deveria ser -1, no entanto, d[a] = 1.