Centro Universitário Presidente Antônio Carlos

Teoria de Grafos

Caminho Mínimo - Algoritmo Bellman-Ford

Felipe Roncalli de Paula Carneiro
felipecarneiro@unipac.br

O que vamos aprender nessa aula

- Grafos com pesos;
- Caminhos Mínimos
- Algoritmo de Bellman-Ford;

Algoritmo de Bellman-Ford

- Alguns autores denominam o algoritmo de Ford-Moore-Bellman, em homenagem a outros três autores que propuseram o mesmo algoritmo em anos diferentes:
- Lester Ford (1956);
- Edward Moore (1957);
- Richard Bellman (1958).

Arestas com Pesos Negativos

Para além das distâncias geográficas, caminhos mais curtos podem modelar diversas outras situações reais, incluindo aquelas que para serem modeladas necessitam de arestas cujo peso é negativo:

- Movimentações financeiras, nas quais é possível obter lucro ou prejuízo, principalmente quando há utilização de câmbio;
- Um taxista que recebe mais dinheiro do que gasta com combustível a cada viagem: se o táxi roda vazio, ele gasta mais do que recebe;
- Um entregador que necessita atravessar um pedágio e pode acabar pagando mais do que recebe para entregar encomendas;
- A energia gerada e consumida durante uma reação química.

Algoritmo de Bellman-Ford

- Ao invés de fechar um vértice por iteração, como o algoritmo de Dijkstra, examina todos os vértices de um grafo orientado por iteração até que atualizações não sejam possíveis.

- Em um grafo com n vértices, qualquer caminho possui no máximo n - 1 arestas, portanto, cada vértice é examinado no máximo n - 1 vezes.

- Com esta estratégia, é possível calcular caminhos mínimos em grafos com arestas de peso negativo.

Algoritmo de Bellman-Ford

- Assim como o algoritmo de Dijkstra, baseia-se no princípio de relaxação: uma aproximação da distância da origem até cada vértice é gradualmente atualizada por valores mais precisos até que a solução ótima seja atingida.

- Se, em alguma iteração do algoritmo os caminhos até cada um dos vértices permanecerem inalterados, não haverá atualizações nas próximas iterações e o algoritmo pode terminar.

Algoritmo de Bellman-Ford

Terminologia

- $\Gamma^{-}(i)$: Conjunto de vértices antecessores do vértice atual;
- $dt[i]$: Vetor que armazena a distância entre o vértice de origem e o vértice i;
- $rot[i]$: Vetor que armazena o índice do vértice anterior ao vértice i, no caminho cuja distância está armazenada em $dt[i]$;
- $altera$: variável *booleana* que indica se houve alguma atualização na iteração atual.

Algoritmo de Bellman-Ford

Entrada: Grafo $G = (V, E)$ e matriz de pesos D=$\{d_{ij}\}$ para todos os arcos (i, j)

$dt[1] \leftarrow 0$; $rot[1] \leftarrow \infty$; //considerando o vértice 1 como o inicial

para $i \leftarrow 2$ **até** n **faça**

  se $\exists$ $(1, i) \in E$ **então** $rot[i] \leftarrow 1$; $dt[i] \leftarrow d_{1i}$;

  senão $rot[i] \leftarrow 0$; $dt[i] \leftarrow \infty$;

fim

para $k \leftarrow 1$ **até** n-1 **faça**

  altera $\leftarrow$ falso;

  para $i \leftarrow 2$ **até** n **faça**

    para $j \in \Gamma^{-}(i)$ **faça**

      se $dt[i] > dt[j]+d_{ji}$ **então**

        $dt[i] \leftarrow dt[j]+d_{ji}$;

        $rot[i] \leftarrow j$;

        altera $\leftarrow$ verdadeiro; //indica que houve alteração

      fim

    fim

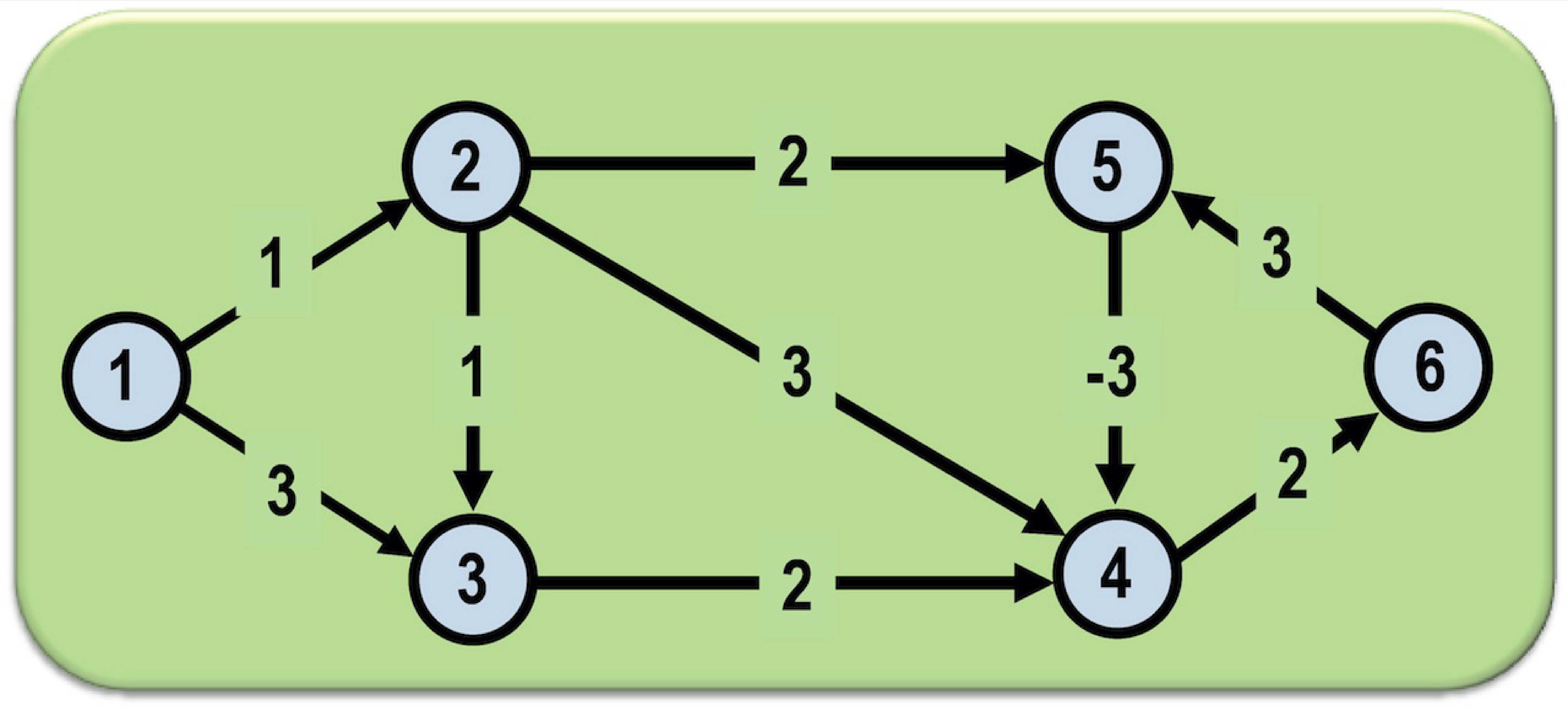
  fim

  se *altera* = *falso* **então** $k \leftarrow n$;

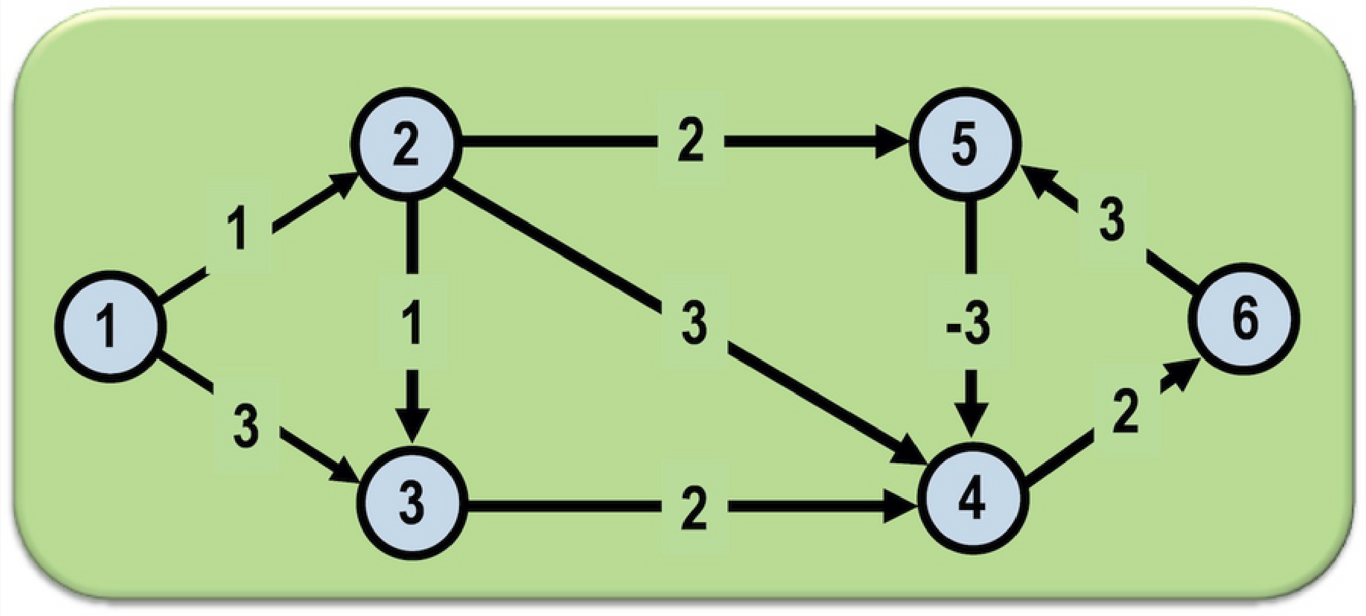
Complexidade

- Após a inicialização, o laço para da linha 6 é repetido por no máximo n-1 vezes;
- Em cada iteração, são calculados caminhos com k arestas entre a origem e os demais vértices do grafo;
- Para cada um dos n-1 vértices, todos seus antecessores são
- examinados;
- O vértice original não é atualizado, logo, n-2 antecessores são analisados no máximo;
- Logo, em uma implementação simples, a complexidade é $O(n^3)$.

Exemplo



Exemplo



dt

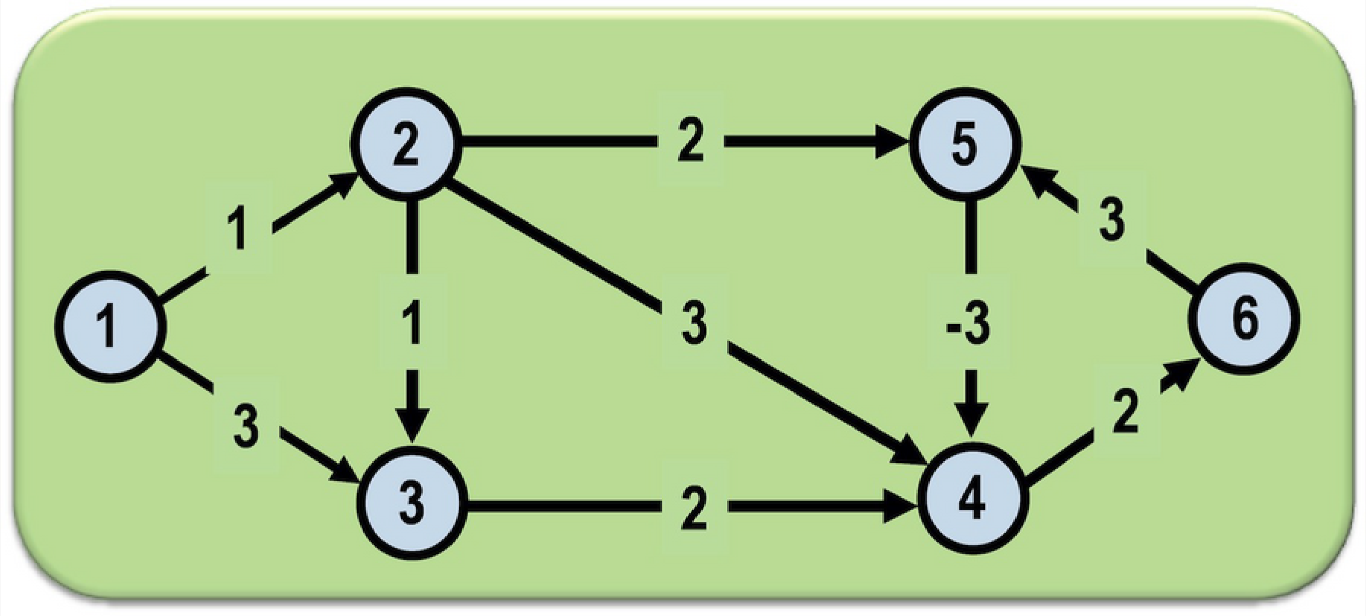
2	3	4	5	6
1	3	∞	∞	∞

rot

2	3	4	5	6
1	1	0	0	0

Vetores após a inicialização do algoritmo.

Exemplo



$i=2,\ \Gamma^{-}(i)=\{1\};$

- $j=1,\ dt[1]+d_{12}=1$

dt

2	3	4	5	6
1	3	∞	∞	∞

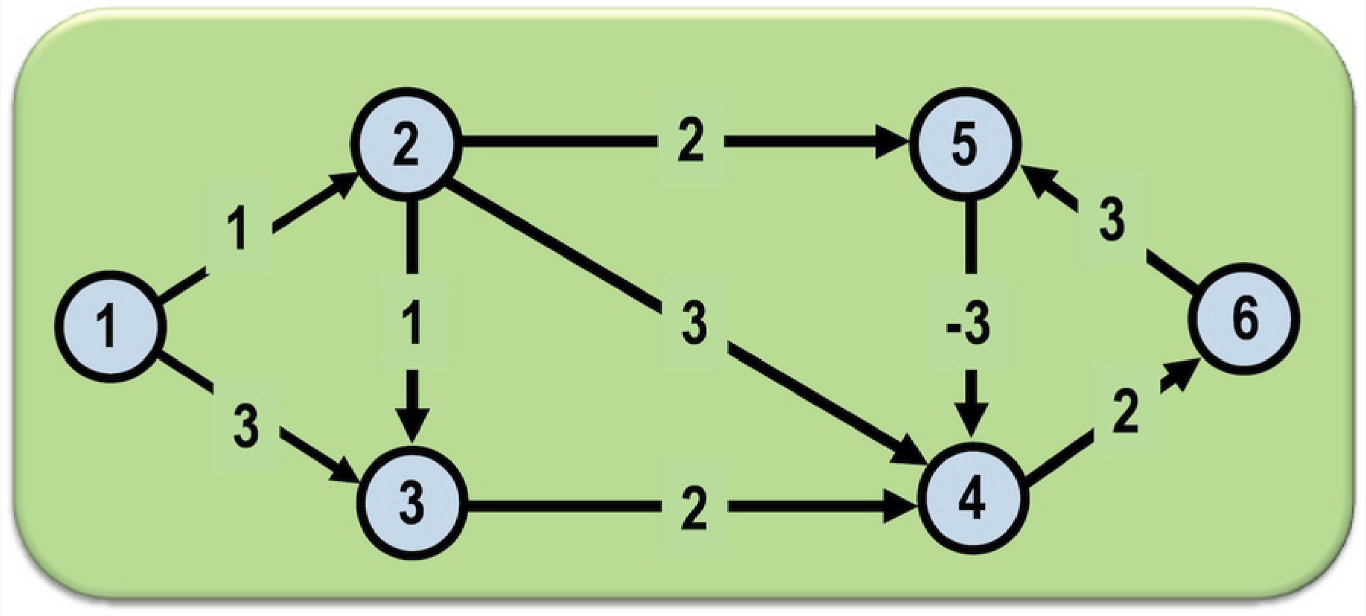
rot

2	3	4	5	6
1	1	0	0	0

Iteração k=1

Valor de i = 2

Exemplo



i=3, $\Gamma^-(i)$={1, 2};

- j=1, $dt[1]+d_{13} = 3$
- j=2, $dt[2]+d_{23} = 2$

dt

2	3	4	5	6
1	2	∞	∞	∞

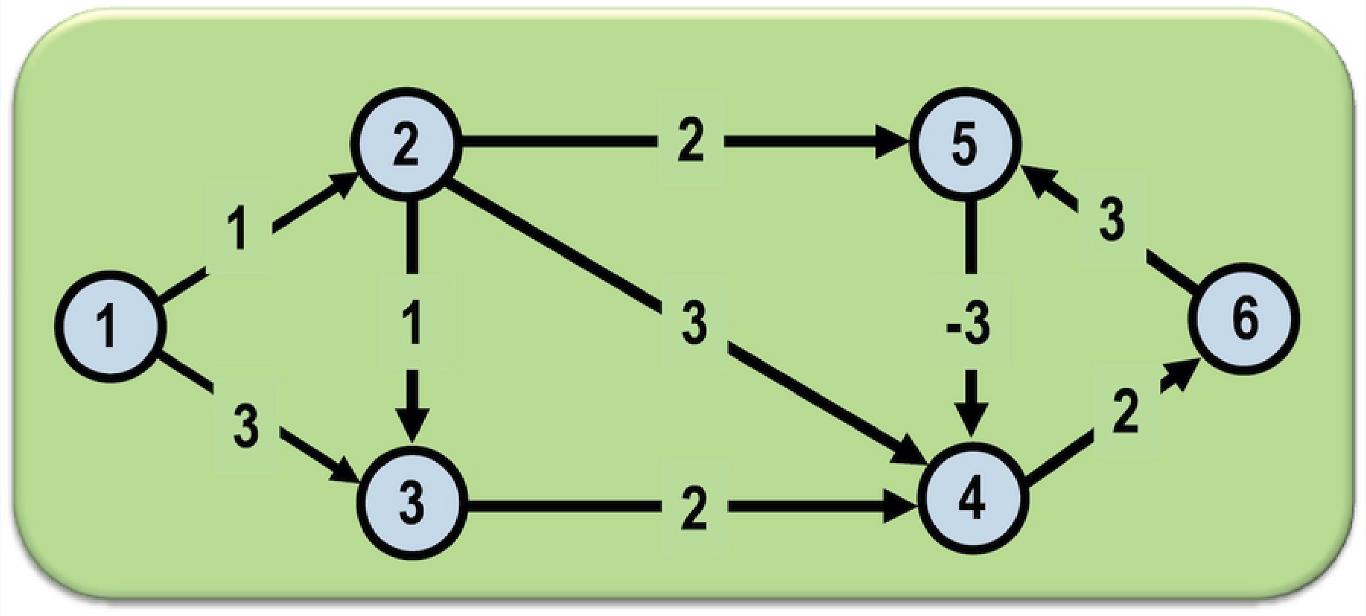
rot

2	3	4	5	6
1	2	0	0	0

Iteração k=1

Valor de i = 3

Exemplo



i=4, $\Gamma^-(i)$={2, 3, 5};

- j=2, dt[2]+d_{24} = 4
- j=3, dt[3]+d_{34} = 4
- j=5, dt[5]+d_{54} = ∞

dt

2	3	4	5	6
1	2	4	∞	∞

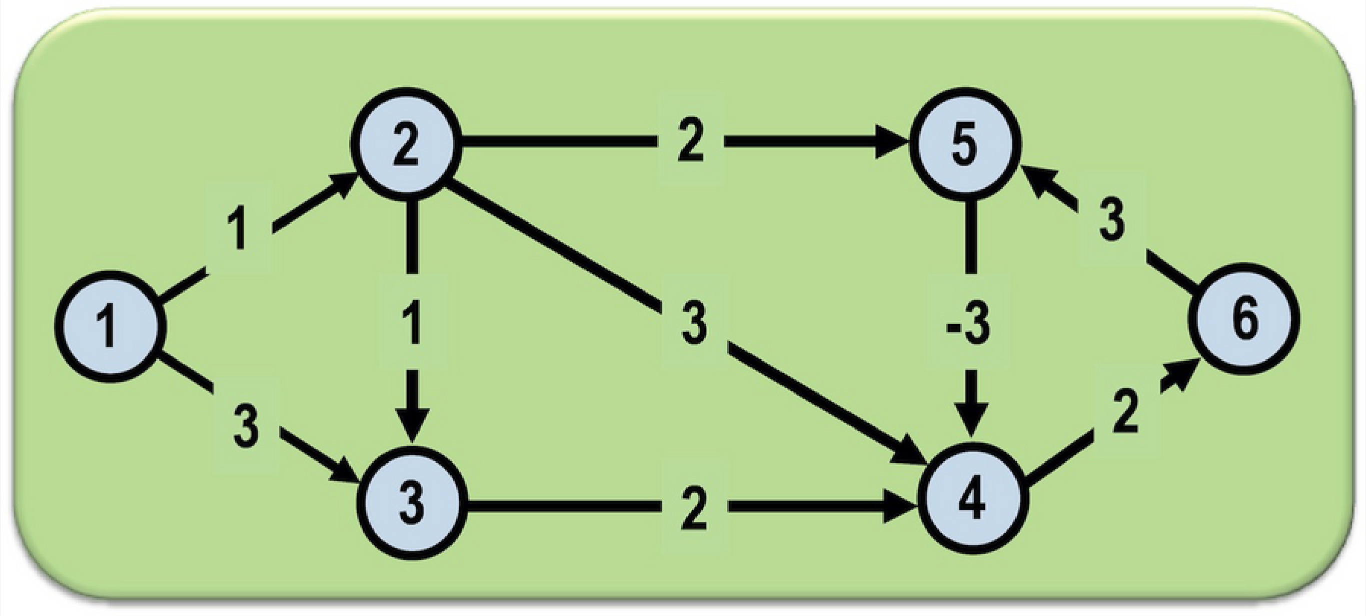
rot

2	3	4	5	6
1	2	2	0	0

Iteração k=1

Valor de i = 4

Exemplo



i=5, $\Gamma^{-}(i)$={2, 6};

- j=2, $dt[2]+d_{25} = 3$
- j=6, $dt[6]+d_{65} = \infty$

dt

2	3	4	5	6
1	2	4	3	∞

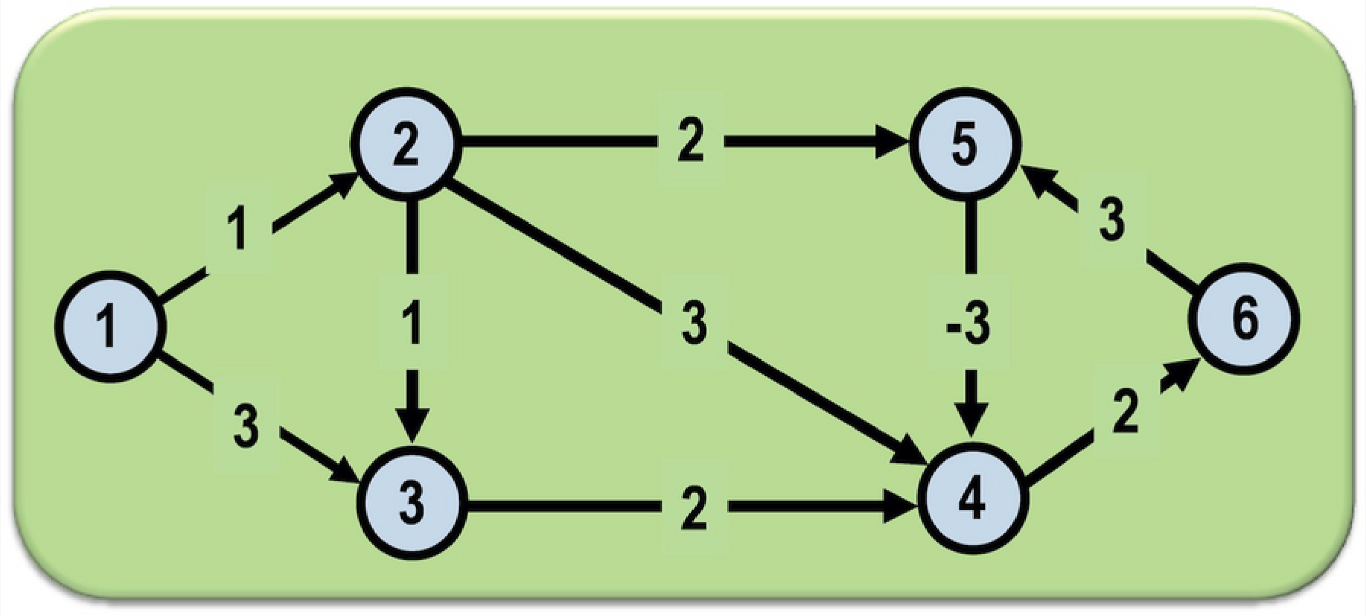
rot

2	3	4	5	6
1	2	2	2	0

Iteração k=1

Valor de i = 5

Exemplo



i=6, $\Gamma^{-}(i)$={4};

- j=4, $dt[4]+d_{46}$ = 6

dt

2	3	4	5	6
1	2	4	3	6

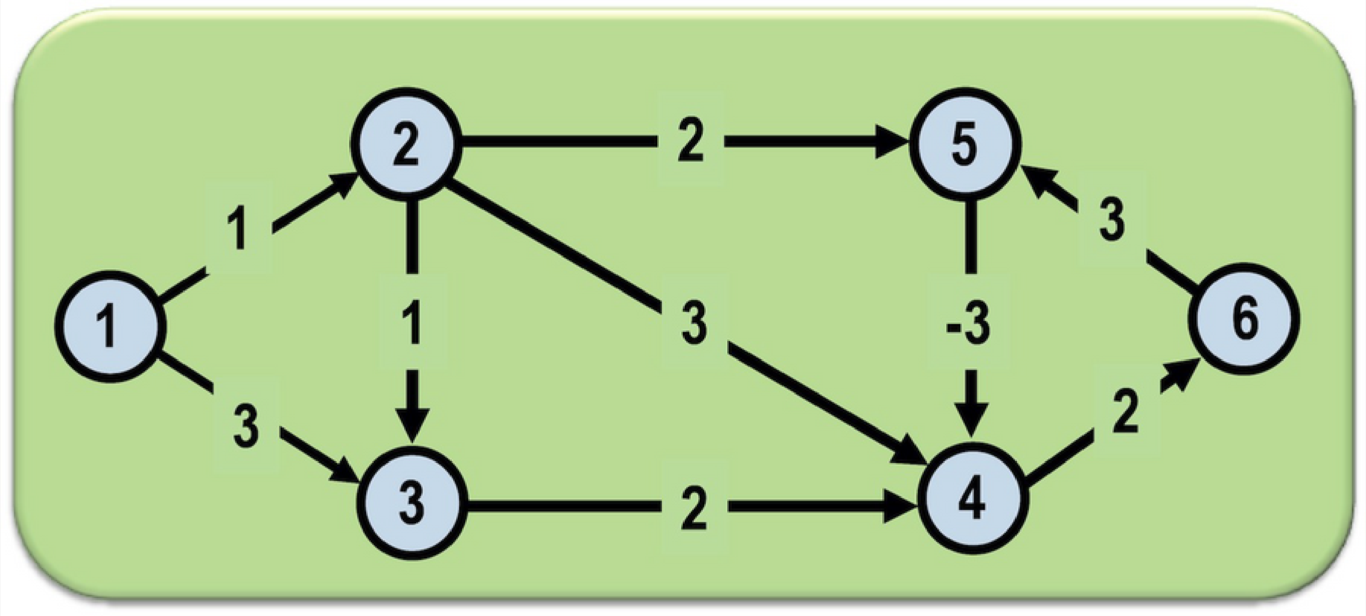
rot

2	3	4	5	6
1	2	2	2	4

Iteração k=1

Valor de i = 6

Exemplo



i=2, $\Gamma^{-}(i)$={1};

- j=1, $dt[1]+d_{12} = 1$

dt

2	3	4	5	6
1	2	4	3	6

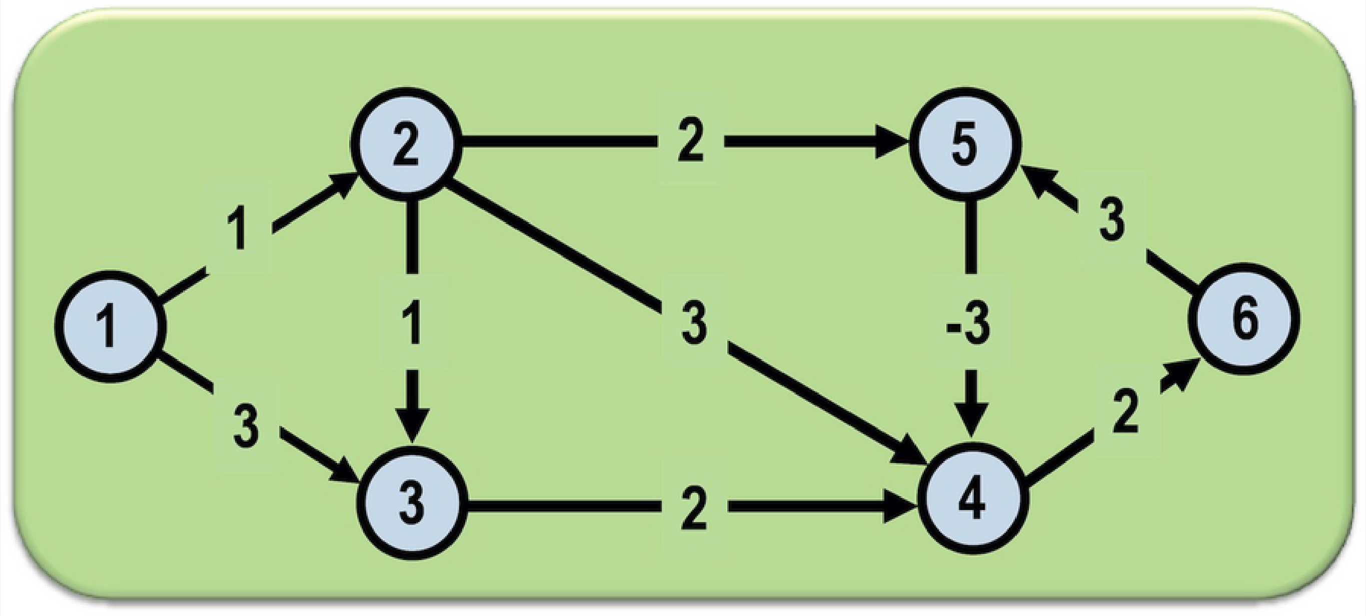
rot

2	3	4	5	6
1	2	2	2	4

Iteração k=2

Valor de i = 2

Exemplo



i=3, $\Gamma^{-}(i)$={1, 2};

- j=1, $dt[1]+d_{13}$ = 3
- j=2, $dt[2]+d_{23}$ = 2

dt

2	3	4	5	6
1	2	4	3	6

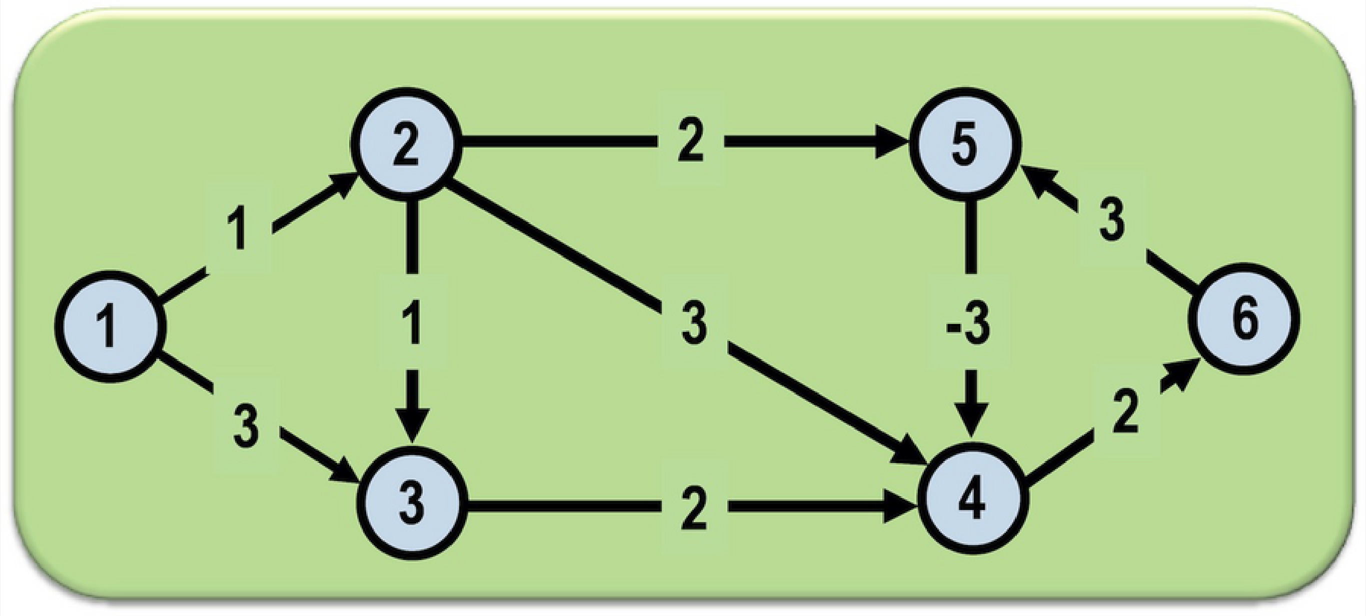
rot

2	3	4	5	6
1	2	2	2	4

Iteração k=2

Valor de i =3

Exemplo



i=4, $\Gamma^{-}(i)$={2, 3, 5};

- j=2, $dt[2]+d_{24}$ = 4
- j=3, $dt[3]+d_{34}$ = 4
- j=5, $dt[5]+d_{54}$ = 0

dt

2	3	4	5	6
1	2	0	3	6

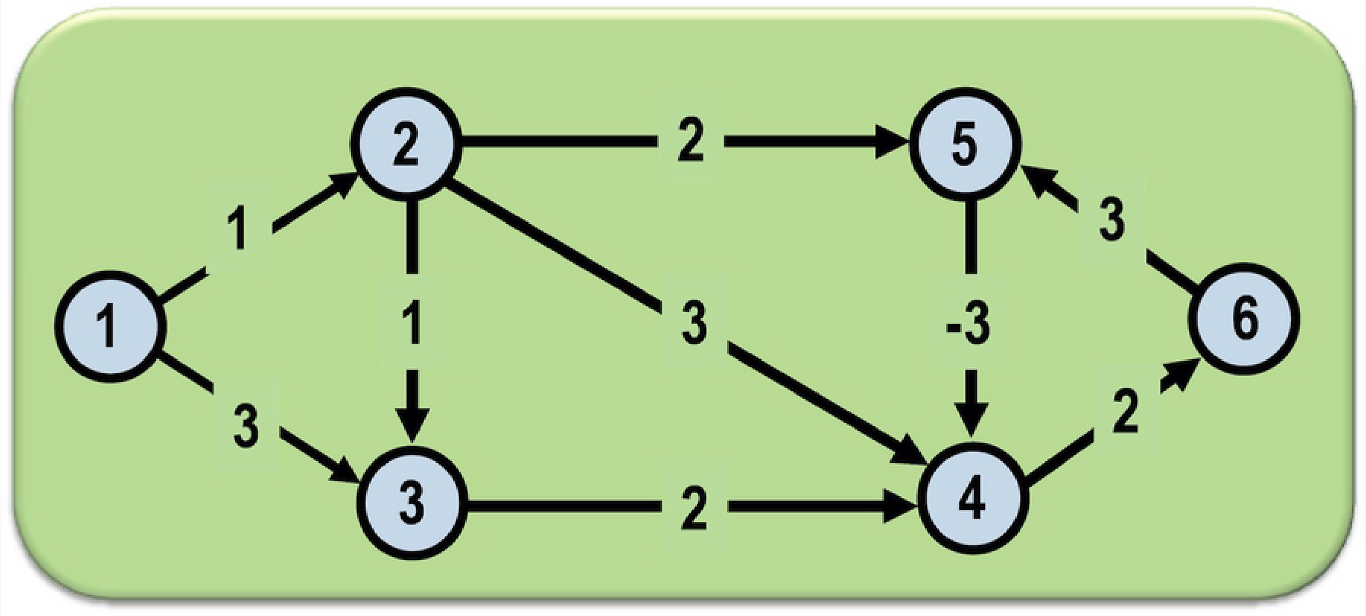
rot

2	3	4	5	6
1	2	5	2	4

Iteração k=2

Valor de i =4

Exemplo



i=5, $\Gamma^-(i)$={2, 6};

- j=2, $dt[2]+d_{25} = 3$
- j=6, $dt[6]+d_{65} = 9$

dt

2	3	4	5	6
1	2	0	3	6

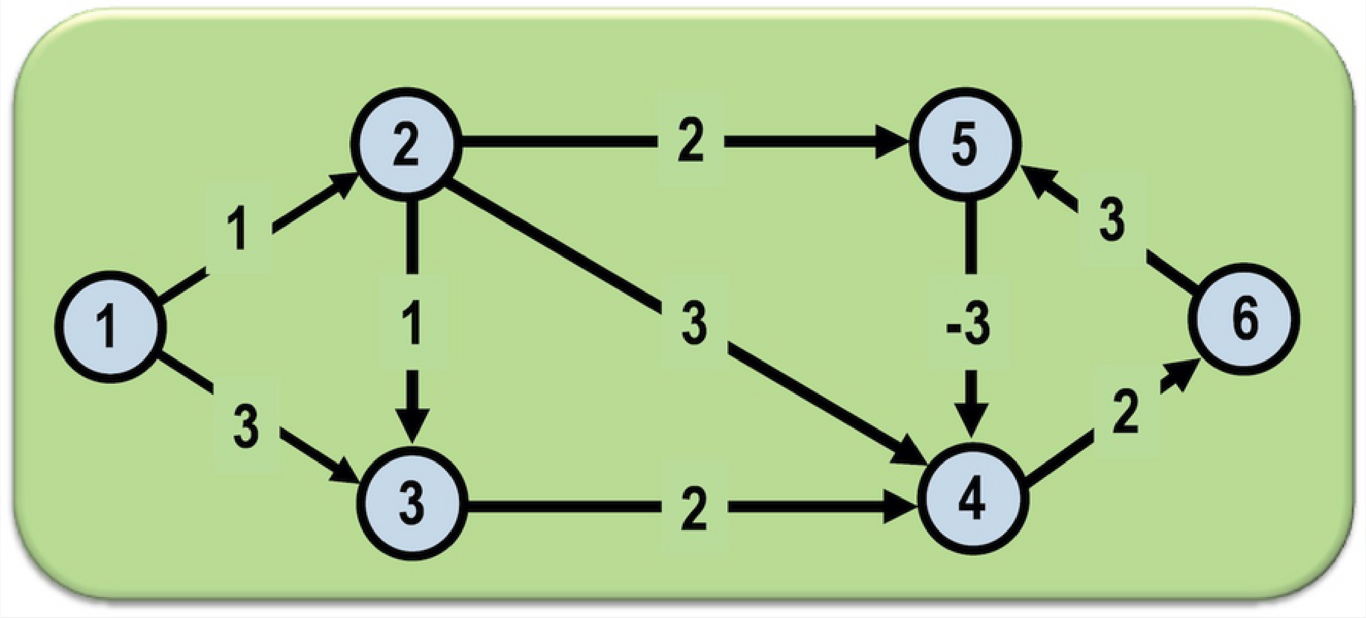
rot

2	3	4	5	6
1	2	5	2	4

Iteração k=2

Valor de i =5

Exemplo



i=6, $\Gamma^{-}(i)$={4};

- j=4, $dt[4]+d_{46}$ = 2

dt

2	3	4	5	6
1	2	0	3	2

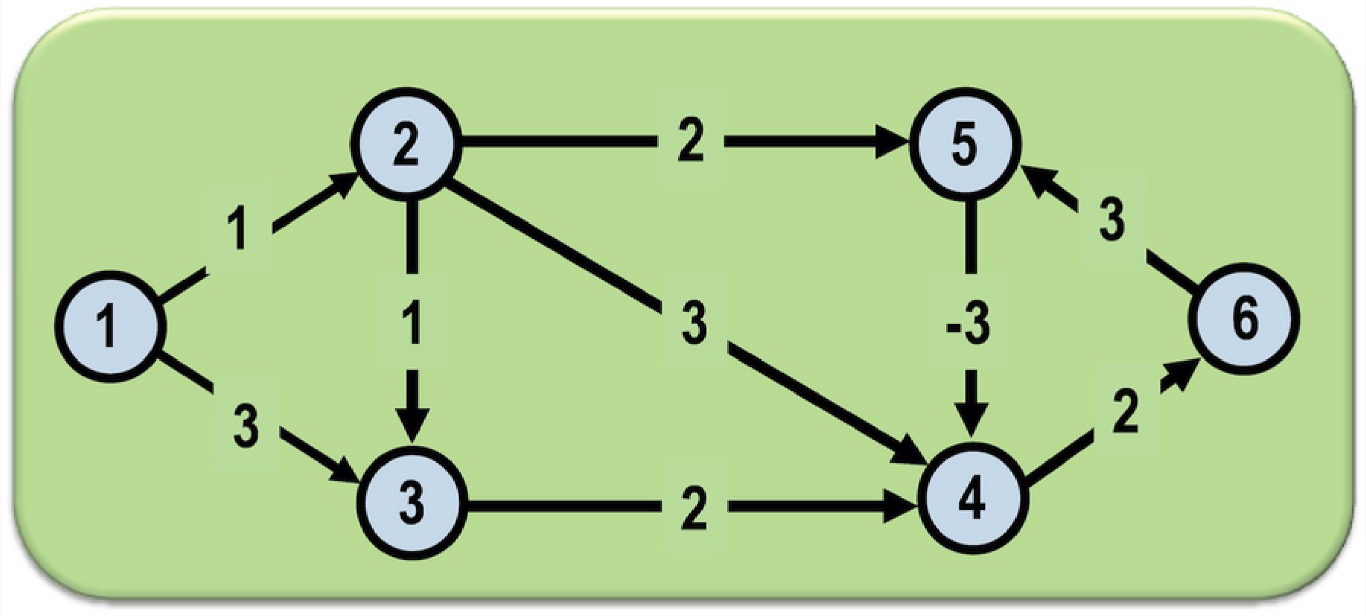
rot

2	3	4	5	6
1	2	5	2	4

Iteração k=2

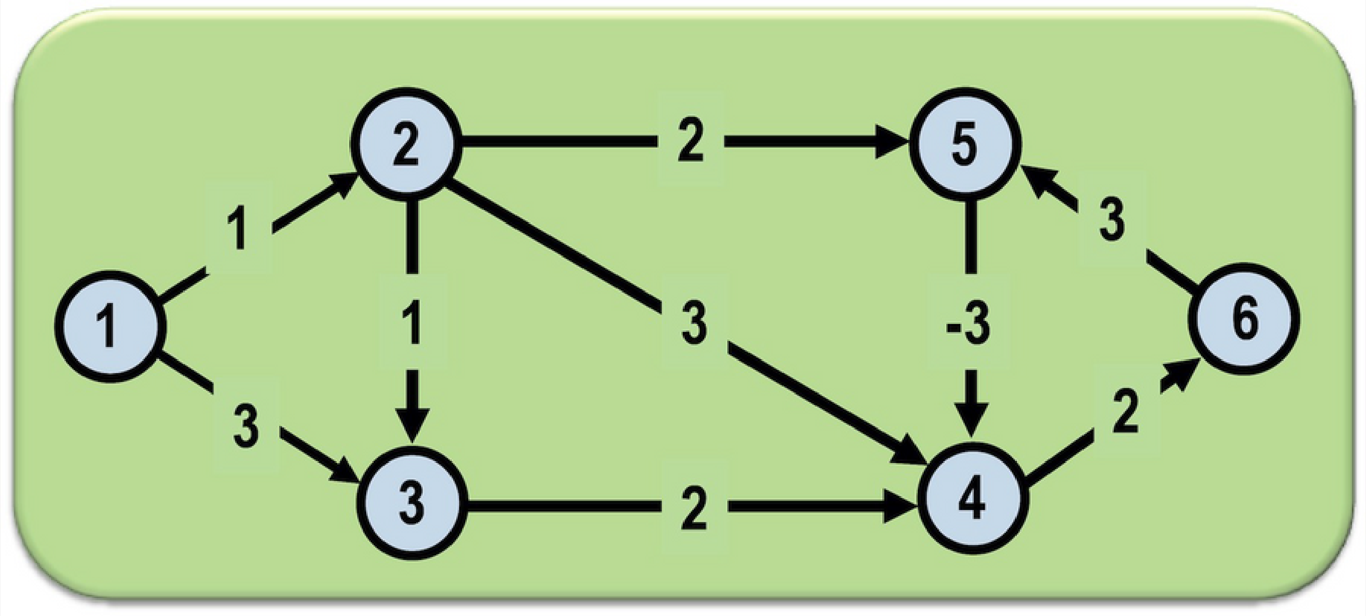
Valor de i =6

Exemplo



Iteração k=3, Teremos Alterações?

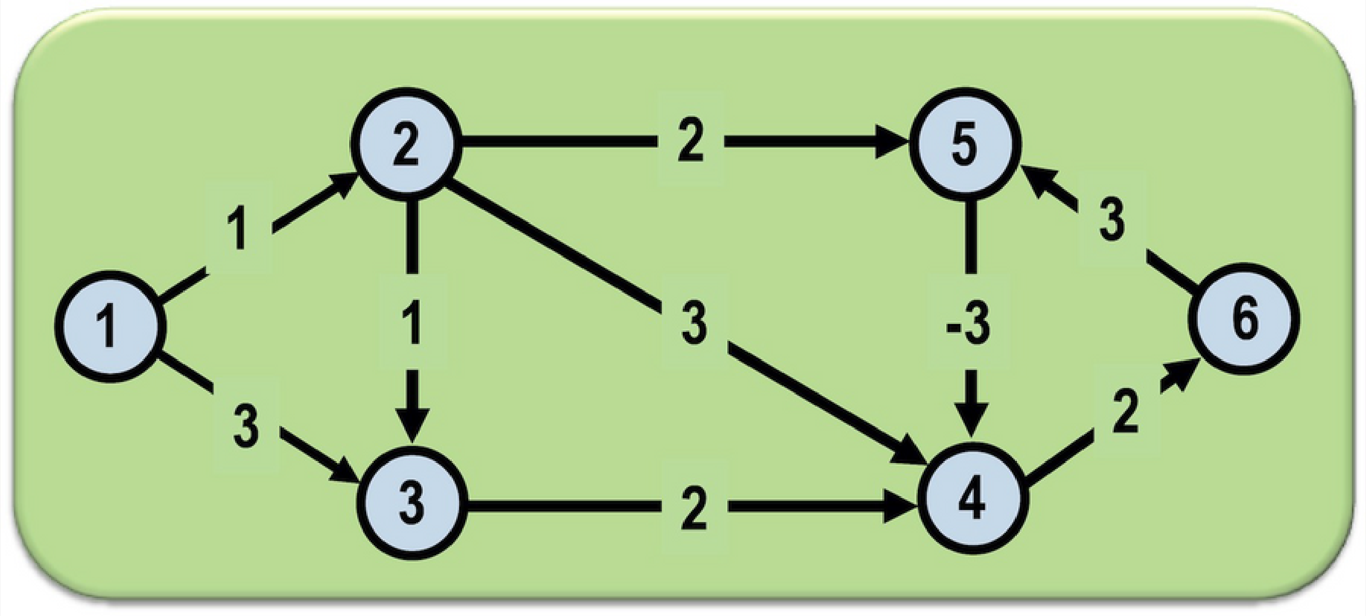
Exemplo



Iteração k=3, Teremos Alterações?

Não Teremos, assim a execução do Algoritmo é encerrada.

Exemplo - Final



dt

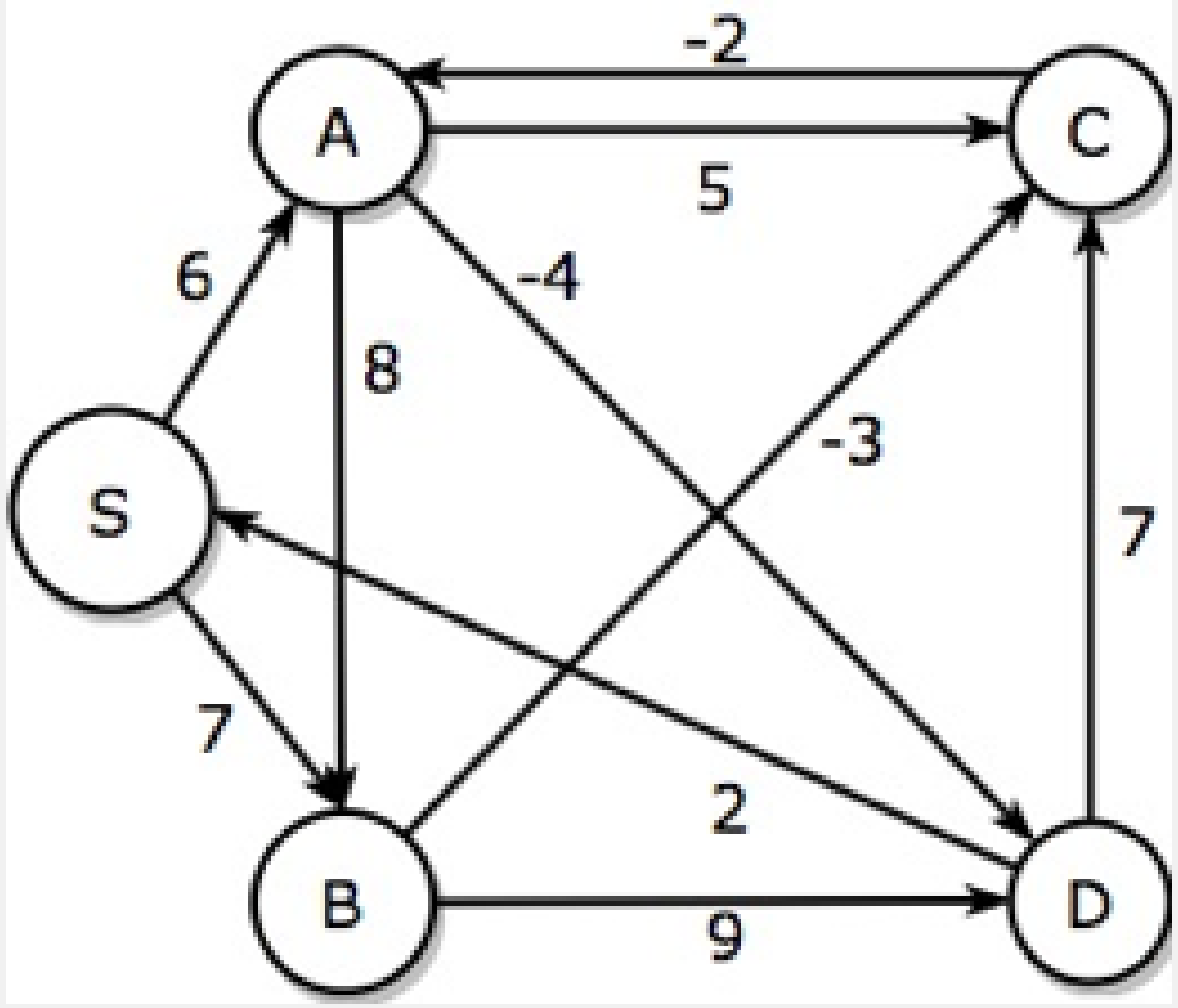
2	3	4	5	6
1	2	0	3	2

rot

2	3	4	5	6
1	2	5	2	4

Exercício

- Encontre o menor caminho utilizando Bellman-Ford partindo de S. Faça o passo a passo.



Dúvidas??