PCC173/BCC463 - Otimização em Redes

Marco Antonio M. Carvalho

Departamento de Computação
Instituto de Ciências Exatas e Biológicas
Universidade Federal de Ouro Preto

Conteúdo

Aviso

Fonte

Este material é baseado nos livros

- Goldbarg, M., & Goldbarg, E. (2012). *Grafos: conceitos, algoritmos e aplicações*. Elsevier.
- Goldbarg, M. C., & Luna, H. P. L. (2005). *Otimização combinatória e programação linear: modelos e algoritmos*. Elsevier.

Licença

Algoritmo de *Floyd-Warshall*

Histórico

O algoritmo proposto por *Robert Floyd* em 1962 é baseado no algoritmo de *Stephen Warshall* do mesmo ano para cálculo de fechos transitivos em grafos.

Seguindo a lei de Stigler para eponímia, o método foi publicado anteriormente em 1959 por *Bernard Roy*.

Referências

Roy, Bernard (1959). "Transitivité et connexité". C. R. Acad. Sci. Paris 249: 216–218.

Warshall, Stephen (January 1962). "A theorem on Boolean matrices". Journal of the ACM 9 (1): 11–12. doi:10.1145/321105.321107.

Floyd, Robert W. (June 1962). "Algorithm 97: Shortest Path". Communications of the ACM 5 (6): 345. doi:10.1145/367766.368168.

Algoritmo de *Floyd-Warshall*

Princípio

O algoritmo de *Floyd-Warshall* calcula os caminhos mais curtos entre todos os pares de vértices de um grafo direcionado e ponderado que eventualmente possua arcos com peso negativo, mas que não possua ciclos de custo negativo.

Trata-se novamente de um algoritmo de programação dinâmica *bottom-up*.

Também de maneira similar, o conceito de relaxação do comprimento dos caminhos mais curtos é empregado: incrementalmente, aprimora-se uma estimativa utilizada, até que o valor ótimo seja atingido.

Algoritmo de *Floyd-Warshall*

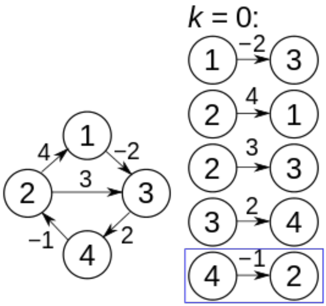
Princípio

O algoritmo compara os caminhos entre os vértices i e j passando por k vértices intermediários, $k = 1, \ldots, n$.

Em outras palavras, todos os caminhos entre cada par de vértices são analisados.

Uma matriz armazena o valor dos caminhos mais curtos entre os vértices, porém, não há informação sobre composição do caminho.

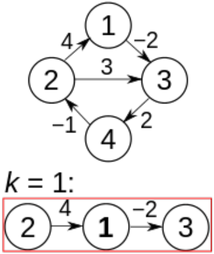
Exemplo Compacto



k=0

Na iteração k=0, somente os caminhos representados por uma única adjacência no grafo são conhecidos.

Exemplo Compacto

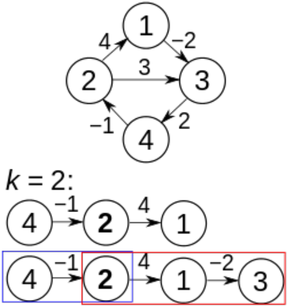


k=1

Na iteração iteração k=1, todos os caminhos que passam pelo vértice 1 são descobertos.

Em particular, o caminho [2,1,3] substitui o caminho [2,3].

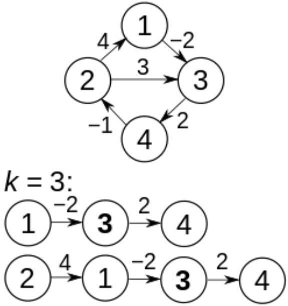
Exemplo Compacto



k=2

Na iteração iteração k=2, todos os caminhos que passam pelos vértices 1 e 2 são descobertos.

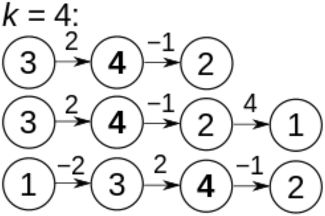
Em particular, o caminho [4, 2, 3] não é considerado, dado que [4, 2, 1, 3] é um caminho mais curto até então.



k=3

Na iteração k=3, todos os caminhos que passam pelos vértices 1, 2 e 3 são descobertos.

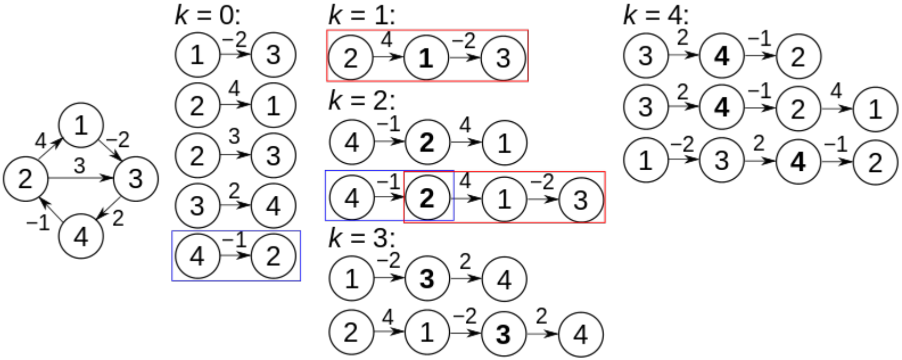
Exemplo Compacto



k=4

Finalmente, em k=4, todos os caminhos mais curtos são determinados.

Exemplo Compacto



Algoritmo de *Floyd-Warshall*

Terminologia

- L: Matriz que armazena os caminhos mais curtos entre os vértices;
  - Inicialmente, L é inicializada com os pesos dos arcos do grafo (d_{ij});
  - Caso não haja arco entre dois vértices i e j, $d_{ij} = \infty$.
- l_{ij}: elemento da matriz L na linha i e coluna j.

Referência

A versão do algoritmo com três laços aninhados é devida a *Peter Ingerman*.

Ingerman, Peter Z. (November 1962). "Algorithm 141: Path Matrix". Communications of the ACM 5 (11): 556. doi:10.1145/368996.369016

Algoritmo de *Floyd-Warshall*

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

$L \leftarrow D$;//Inicializa os elementos da matriz L

para $k \leftarrow 1$ **até** n **faça**
    para $i \leftarrow 1$ **até** n **faça**
        para $j \leftarrow 1$ **até** n **faça**
            se $l_{ij} > l_{ik} + l_{kj}$ **então**
                $l_{ij} \leftarrow l_{ik} + l_{kj}$;
            fim
        fim
    fim
fim

Ciclos de Custo Negativo

O algoritmo de *Floyd-Warshall* detecta ciclos de custo negativo.

Caso haja valores negativos na diagonal principal da matriz L (inclusive durante a execução do algoritmo), significa que o vértice relacionado está contido em um ciclo de custo negativo.

Em outras palavras, é possível sair do vértice, percorrer parte do grafo e retornar ao vértice inicial com custo negativo.

Algumas versões do algoritmo consideram que não haverá ciclos de custo negativo e definem inicialmente a distância de um vértice para si próprio como $-\infty$, implicando em não haver atualização possível.

Alternativa

Outra forma de entender o algoritmo de *Floyd-Warshall* é usando uma relação de recorrência.

Seja $dist(i, j, k)$ o comprimento do caminho mais curto entre i e j tal que apenas os vértices $\{1, \ldots, k\}$ podem ser usados como intermediários.

$$dist(i, j, k + 1) = min\{dist(i, j, k), dist(i, k + 1, k) + dist(k + 1, j, k)\}$$

O algoritmo calcula a recorrência acima para $k = 1 \ldots n$.

Algoritmo de *Floyd-Warshall*

Complexidade

Os três laços aninhados são executados n vezes, logo, a complexidade final é $O(n^3)$, ou mais precisamente, $\Theta(n^3)$.

Embora a complexidade seja alta, é importante notar que todas as arestas são verificadas, e um grafo pode ter mais do que n^2 arestas, tornando-o uma boa opção para grafos densos.

Para grafos esparsos e com pesos positivos, executar uma boa implementação do algoritmo de Dijkstra para cada um dos n vértices é uma melhor opção, com complexidade $O(nm\ lg\ n)$.

Caso haja arestas de peso negativo, o algoritmo de *Johnson* pode ser utilizado.

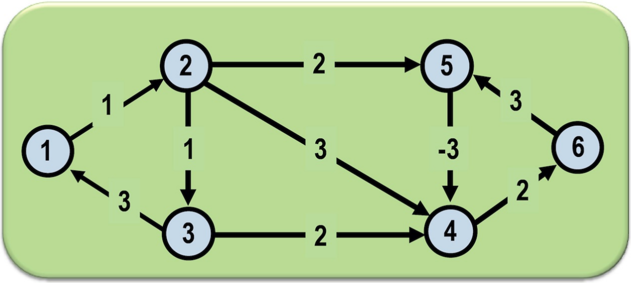
Composição dos Caminhos

Conforme visto, não são armazenadas informações sobre quais vértices compõem os caminhos mais curtos calculados, no entanto, o algoritmo pode ser modificado.

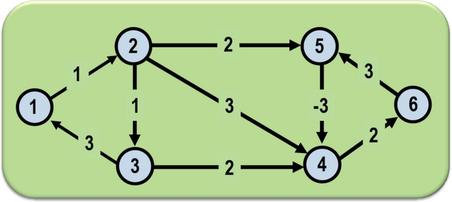
Não é necessário, entretanto, armazenar de fato todos os vértices dos caminhos mais curtos, implicando na necessidade de uma matriz tridimensional.

Árvores de Caminhos Mais Curtos podem ser utilizadas para este fim.

Exemplo Passo a Passo



Exemplo Passo a Passo



	1	2	3	4	5	6
1	0	1	∞	∞	∞	∞
2	∞	0	1	3	2	∞
3	3	∞	0	2	∞	∞
4	∞	∞	∞	0	∞	2
5	∞	∞	∞	-3	0	∞
6	∞	∞	∞	∞	3	0

Matriz L inicial.

Exemplo Passo a Passo

- j=1, l_{11} = min{l_{11}; (l_{11}+l_{11})}=0
- j=2, l_{12} = min{l_{12}; (l_{11}+l_{12})}=1
- j=3, l_{13} = min{l_{13}; (l_{11}+l_{13})}=∞
- j=4, l_{14} = min{l_{14}; (l_{11}+l_{14})}=∞
- j=5, l_{15} = min{l_{15}; (l_{11}+l_{15})}=∞
- j=6, l_{16} = min{l_{16}; (l_{11}+l_{16})}=∞

	1	**2**	**3**	**4**	**5**	**6**
1	0	1	∞	∞	∞	∞
2	∞	0	1	3	2	∞
3	3	∞	0	2	∞	∞
4	∞	∞	∞	0	∞	2
5	∞	∞	∞	-3	0	∞
6	∞	∞	∞	∞	3	0

Iteração k = 1, i=1.
A matriz não é alterada.

Exemplo Passo a Passo

- j=1, l_{21} = min$\{l_{21}; (l_{21}+l_{11})\}=\infty$
- j=2, l_{22} = min$\{l_{22}; (l_{21}+l_{12})\}=0$
- j=3, l_{23} = min$\{l_{23}; (l_{21}+l_{13})\}=1$
- j=4, l_{24} = min$\{l_{24}; (l_{21}+l_{14})\}=3$
- j=5, l_{25} = min$\{l_{25}; (l_{21}+l_{15})\}=2$
- j=6, l_{26} = min$\{l_{26}; (l_{21}+l_{16})\}=\infty$

	1	**2**	**3**	**4**	**5**	**6**
1	0	1	∞	∞	∞	∞
2	∞	0	1	3	2	∞
3	3	∞	0	2	∞	∞
4	∞	∞	∞	0	∞	2
5	∞	∞	∞	-3	0	∞
6	∞	∞	∞	∞	3	0

Iteração k = 1, i=2.
A matriz não é alterada novamente.

Exemplo Passo a Passo

- j=1, l_{31} = min{l_{31}; (l_{31}+l_{11})}=3
- j=2, l_{32} = min{l_{32}; (l_{31}+l_{12})}=4
- j=3, l_{33} = min{l_{33}; (l_{31}+l_{13})}=0
- j=4, l_{34} = min{l_{34}; (l_{31}+l_{14})}=2
- j=5, l_{35} = min{l_{35}; (l_{31}+l_{15})}=∞
- j=6, l_{36} = min{l_{36}; (l_{31}+l_{16})}=∞

	1	**2**	**3**	**4**	**5**	**6**
1	0	1	∞	∞	∞	∞
2	∞	0	1	3	2	∞
3	3	∞	0	2	∞	∞
4	∞	∞	∞	0	∞	2
5	∞	∞	∞	-3	0	∞
6	∞	∞	∞	∞	3	0

Iteração k = 1, i=3.
A matriz é alterada!

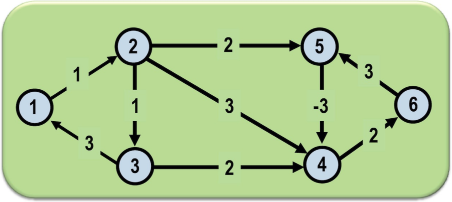
Fast Forward...

Não há outras alterações para k=1 e i=4, 5, 6.

Nos *slides* a seguir só serão exibidas as iterações do algoritmo em que ocorrem alterações na matriz L.

Atenção: este exemplo possui erros de digitação nos livros do Goldbarg!

Exemplo Passo a Passo



	1	2	3	4	5	6
1	0	1	∞	∞	∞	∞
2	∞	0	1	3	2	∞
3	3	4	0	2	∞	∞
4	∞	∞	∞	0	∞	2
5	∞	∞	∞	-3	0	∞
6	∞	∞	∞	∞	3	0

Matriz L para $k = 1$.

Exemplo Passo a Passo

- j=1, l_{11} = min{l_{11}; (l_{12}+l_{21})}=0
- j=2, l_{12} = min{l_{12}; (l_{12}+l_{22})}=1
- j=3, l_{13} = min{l_{13}; (l_{12}+l_{23})}=2
- j=4, l_{14} = min{l_{14}; (l_{12}+l_{24})}=4
- j=5, l_{15} = min{l_{15}; (l_{12}+l_{25})}=3
- j=6, l_{16} = min{l_{16}; (l_{12}+l_{26})}=∞

	1	**2**	**3**	**4**	**5**	**6**
1	0	1	∞	∞	∞	∞
2	∞	0	1	3	2	∞
3	3	4	0	2	∞	∞
4	∞	∞	∞	0	∞	2
5	∞	∞	∞	-3	0	∞
6	∞	∞	∞	∞	3	0

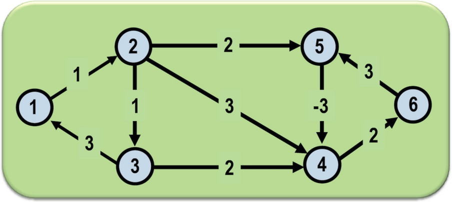
Iteração k = 2, i=1.

- j=1, l_{31} = min$\{l_{31}$; $(l_{32}+l_{21})\}$=3
- j=2, l_{32} = min$\{l_{32}$; $(l_{32}+l_{22})\}$=4
- j=3, l_{33} = min$\{l_{33}$; $(l_{32}+l_{23})\}$=0
- j=4, l_{34} = min$\{l_{34}$; $(l_{32}+l_{24})\}$=2
- j=5, l_{35} = min$\{l_{35}$; $(l_{32}+l_{25})\}$=6
- j=6, l_{36} = min$\{l_{36}$; $(l_{32}+l_{26})\}$=∞

	1	**2**	**3**	**4**	**5**	**6**
1	0	1	2	4	3	∞
2	∞	0	1	3	2	∞
3	3	4	0	2	∞	∞
4	∞	∞	∞	0	∞	2
5	∞	∞	∞	-3	0	∞
6	∞	∞	∞	∞	3	0

Iteração k = 2, i=3.

Exemplo Passo a Passo



	1	2	3	4	5	6
1	0	1	2	4	3	∞
2	∞	0	1	3	2	∞
3	3	4	0	2	6	∞
4	∞	∞	∞	0	∞	2
5	∞	∞	∞	-3	0	∞
6	∞	∞	∞	∞	3	0

Matriz L para $k = 2$.

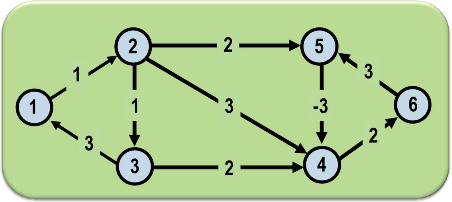
Exemplo Passo a Passo

- j=1, l_{21} = min{l_{21}; (l_{23}+l_{31})}=4
- j=2, l_{22} = min{l_{22}; (l_{23}+l_{32})}=0
- j=3, l_{23} = min{l_{23}; (l_{23}+l_{33})}=1
- j=4, l_{24} = min{l_{24}; (l_{23}+l_{34})}=3
- j=5, l_{25} = min{l_{25}; (l_{23}+l_{35})}=2
- j=6, l_{26} = min{l_{26}; (l_{23}+l_{36})}=∞

	1	**2**	**3**	**4**	**5**	**6**
1	0	1	2	4	3	∞
2	∞	0	1	3	2	∞
3	3	4	0	2	6	∞
4	∞	∞	∞	0	∞	2
5	∞	∞	∞	-3	0	∞
6	∞	∞	∞	∞	3	0

Iteração k = 3, i=2.

Exemplo Passo a Passo



	1	2	3	4	5	6
1	0	1	2	4	3	∞
2	4	0	1	3	2	∞
3	3	4	0	2	6	∞
4	∞	∞	∞	0	∞	2
5	∞	∞	∞	-3	0	∞
6	∞	∞	∞	∞	3	0

Matriz L para $k = 3$.

- j=1, l_{11} = min{l_{11}; (l_{14}+l_{41})}=0
- j=2, l_{12} = min{l_{12}; (l_{14}+l_{42})}=1
- j=3, l_{13} = min{l_{13}; (l_{14}+l_{43})}=2
- j=4, l_{14} = min{l_{14}; (l_{14}+l_{44})}=4
- j=5, l_{15} = min{l_{15}; (l_{14}+l_{45})}=3
- j=6, l_{16} = min{l_{16}; (l_{14}+l_{46})}=6

	1	**2**	**3**	**4**	**5**	**6**
1	0	1	2	4	3	∞
2	4	0	1	3	2	∞
3	3	4	0	2	6	∞
4	∞	∞	∞	0	∞	2
5	∞	∞	∞	-3	0	∞
6	∞	∞	∞	∞	3	0

Iteração k = 4, i=1.

Exemplo Passo a Passo

- j=1, l_{21} = min{l_{21}; (l_{24}+l_{41})}=4
- j=2, l_{22} = min{l_{22}; (l_{24}+l_{42})}=0
- j=3, l_{23} = min{l_{23}; (l_{24}+l_{43})}=1
- j=4, l_{24} = min{l_{24}; (l_{24}+l_{44})}=3
- j=5, l_{25} = min{l_{25}; (l_{24}+l_{45})}=2
- j=6, l_{26} = min{l_{26}; (l_{24}+l_{46})}=5

	1	**2**	**3**	**4**	**5**	**6**
1	0	1	2	4	3	6
2	4	0	1	3	2	∞
3	3	4	0	2	6	∞
4	∞	∞	∞	0	∞	2
5	∞	∞	∞	-3	0	∞
6	∞	∞	∞	∞	3	0

Iteração k = 4, i=2.

Exemplo Passo a Passo

- j=1, l_{31} = min{l_{31}; (l_{34}+l_{41})}=3
- j=2, l_{32} = min{l_{32}; (l_{34}+l_{42})}=4
- j=3, l_{33} = min{l_{33}; (l_{34}+l_{43})}=0
- j=4, l_{34} = min{l_{34}; (l_{34}+l_{44})}=2
- j=5, l_{35} = min{l_{35}; (l_{34}+l_{45})}=6
- j=6, l_{36} = min{l_{36}; (l_{34}+l_{46})}=4

	1	**2**	**3**	**4**	**5**	**6**
1	0	1	2	4	3	6
2	4	0	1	3	2	5
3	3	4	0	2	6	∞
4	∞	∞	∞	0	∞	2
5	∞	∞	∞	-3	0	∞
6	∞	∞	∞	∞	3	0

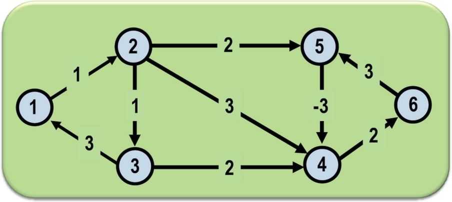
Iteração k = 4, i=3.

- j=1, $l_{51} = \min\{l_{51}; (l_{54}+l_{41})\}=\infty$
- j=2, $l_{52} = \min\{l_{52}; (l_{54}+l_{42})\}=\infty$
- j=3, $l_{53} = \min\{l_{53}; (l_{54}+l_{43})\}=\infty$
- j=4, $l_{54} = \min\{l_{54}; (l_{54}+l_{44})\}$=-3
- j=5, $l_{55} = \min\{l_{55}; (l_{54}+l_{45})\}=0$
- j=6, $l_{56} = \min\{l_{56}; (l_{54}+l_{46})\}$=-1

	1	**2**	**3**	**4**	**5**	**6**
1	0	1	2	4	3	6
2	4	0	1	3	2	5
3	3	4	0	2	6	4
4	∞	∞	∞	0	∞	2
5	∞	∞	∞	-3	0	∞
6	∞	∞	∞	∞	3	0

Iteração k = 4, i=5.

Exemplo Passo a Passo



	1	2	3	4	5	6
1	0	1	2	4	3	6
2	4	0	1	3	2	5
3	3	4	0	2	6	4
4	∞	∞	∞	0	∞	2
5	∞	∞	∞	-3	0	-1
6	∞	∞	∞	∞	3	0

Matriz L para $k = 4$.

- j=1, l_{11} = min{l_{11}; (l_{15}+l_{51})}=0
- j=2, l_{12} = min{l_{12}; (l_{15}+l_{52})}=1
- j=3, l_{13} = min{l_{13}; (l_{15}+l_{53})}=2
- j=4, l_{14} = min{l_{14}; (l_{15}+l_{54})}=0
- j=5, l_{15} = min{l_{15}; (l_{15}+l_{55})}=3
- j=6, l_{16} = min{l_{16}; (l_{15}+l_{56})}=2

	1	**2**	**3**	**4**	**5**	**6**
1	0	1	2	4	3	6
2	4	0	1	3	2	5
3	3	4	0	2	6	4
4	∞	∞	∞	0	∞	2
5	∞	∞	∞	-3	0	-1
6	∞	∞	∞	∞	3	0

Iteração $k = 5$, i=1.

- j=1, l_{21} = min{l_{21}; (l_{25}+l_{51})}=4
- j=2, l_{22} = min{l_{22}; (l_{25}+l_{52})}=0
- j=3, l_{23} = min{l_{23}; (l_{25}+l_{53})}=1
- j=4, l_{24} = min{l_{24}; (l_{25}+l_{54})}=-1
- j=5, l_{25} = min{l_{25}; (l_{25}+l_{55})}=2
- j=6, l_{26} = min{l_{26}; (l_{25}+l_{56})}=1

	1	**2**	**3**	**4**	**5**	**6**
1	0	1	2	0	3	2
2	4	0	1	3	2	5
3	3	4	0	2	6	4
4	∞	∞	∞	0	∞	2
5	∞	∞	∞	-3	0	-1
6	∞	∞	∞	∞	3	0

Iteração k = 5, i=2.

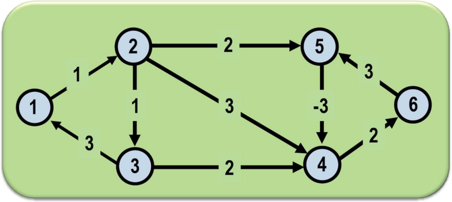
Exemplo Passo a Passo

- j=1, l_{61} = min$\{l_{61}; (l_{65}+l_{51})\}$=$\infty$
- j=2, l_{62} = min$\{l_{62}; (l_{65}+l_{52})\}$=$\infty$
- j=3, l_{63} = min$\{l_{63}; (l_{65}+l_{53})\}$=$\infty$
- j=4, l_{64} = min$\{l_{64}; (l_{65}+l_{54})\}$=0
- j=5, l_{65} = min$\{l_{65}; (l_{65}+l_{55})\}$=3
- j=6, l_{66} = min$\{l_{66}; (l_{65}+l_{56})\}$=0

	1	**2**	**3**	**4**	**5**	**6**
1	0	1	2	0	3	2
2	4	0	1	3	2	5
3	3	4	0	2	6	4
4	∞	∞	∞	0	∞	2
5	∞	∞	∞	-3	0	-1
6	∞	∞	∞	∞	3	0

Iteração k = 5, i=6.

Exemplo Passo a Passo



	1	2	3	4	5	6
1	0	1	2	0	3	2
2	4	0	1	-1	2	1
3	3	4	0	2	6	4
4	∞	∞	∞	0	∞	2
5	∞	∞	∞	-3	0	-1
6	∞	∞	∞	0	3	0

Matriz L para $k = 5$.

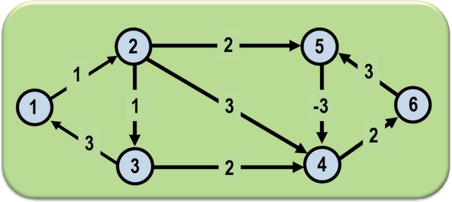
Exemplo Passo a Passo

- j=1, $l_{41} = \min\{l_{41}; (l_{46}+l_{61})\}=\infty$
- j=2, $l_{42} = \min\{l_{42}; (l_{46}+l_{62})\}=\infty$
- j=3, $l_{43} = \min\{l_{43}; (l_{46}+l_{63})\}=\infty$
- j=4, $l_{44} = \min\{l_{44}; (l_{46}+l_{64})\}=0$
- j=5, $l_{45} = \min\{l_{45}; (l_{46}+l_{65})\}=5$
- j=6, $l_{46} = \min\{l_{46}; (l_{46}+l_{66})\}=4$

	1	**2**	**3**	**4**	**5**	**6**
1	0	1	2	0	3	2
2	4	0	1	-1	2	1
3	3	4	0	2	6	4
4	∞	∞	∞	0	∞	2
5	∞	∞	∞	-3	0	-1
6	∞	∞	∞	0	3	0

Iteração k = 6, i=4.

Exemplo Passo a Passo



	1	2	3	4	5	6
1	0	1	2	0	3	2
2	4	0	1	-1	2	1
3	3	4	0	2	6	4
4	∞	∞	∞	0	5	2
5	∞	∞	∞	-3	0	-1
6	∞	∞	∞	0	3	0

Matriz L para $k = 6$.

Dúvidas?