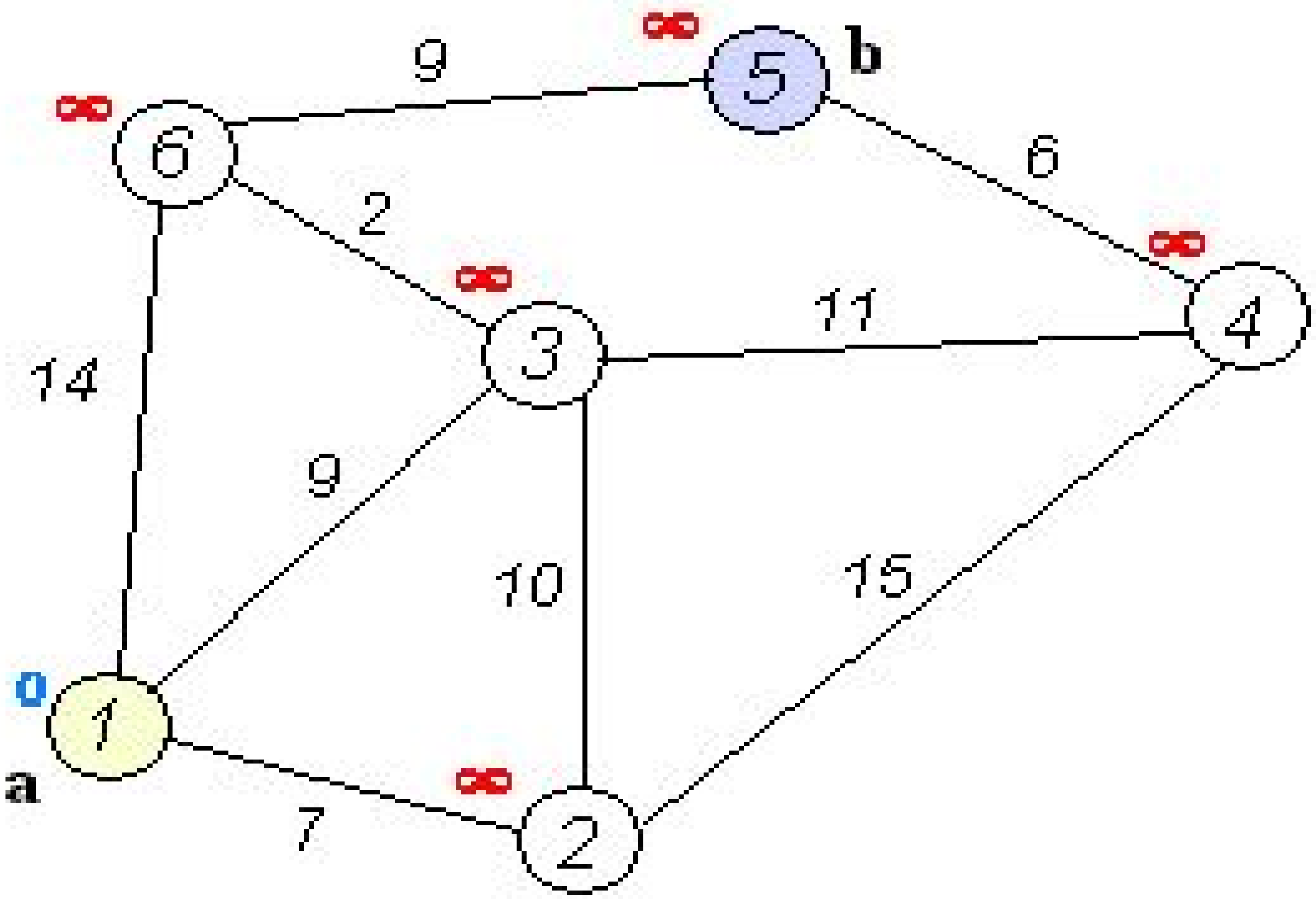
Grafos

Algoritmos e seus funcionamentos

Algoritmo de dijkstra

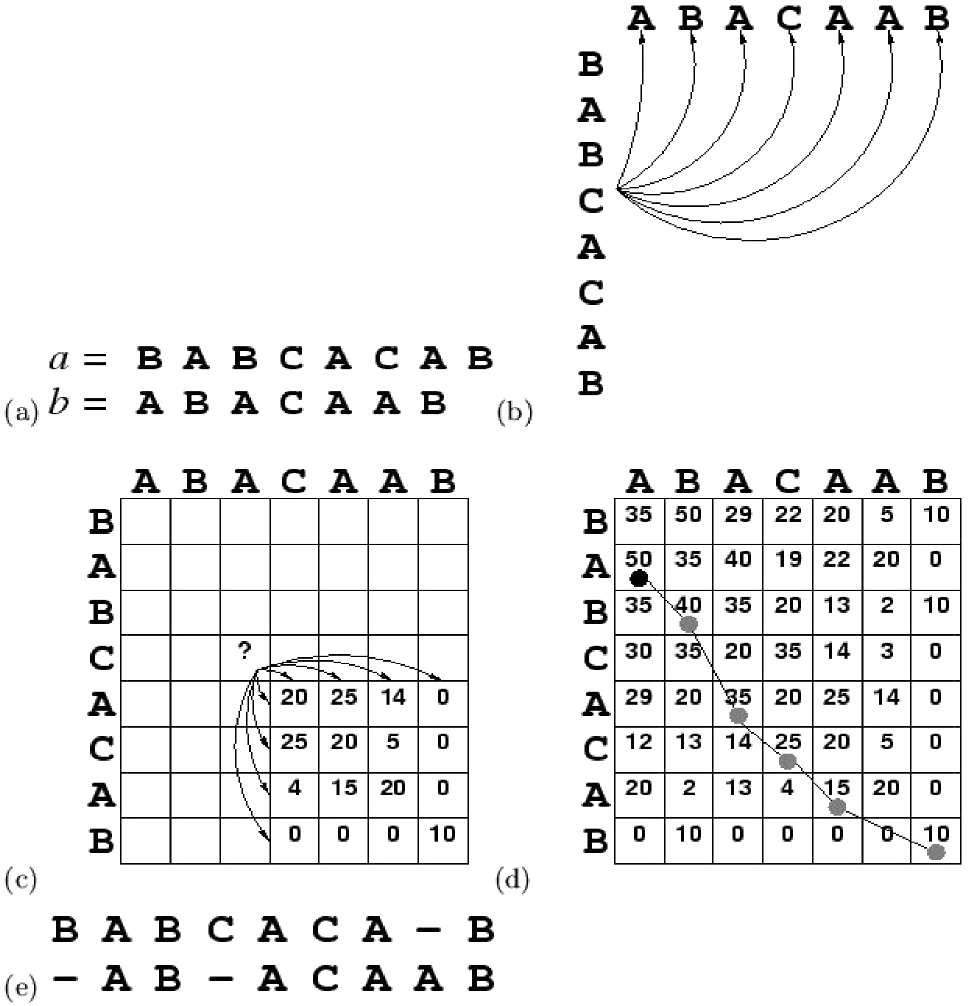
algoritmo de Dijkstra, concebido pelo cientista da computação holandês Edsger Dijkstra em 1956 e publicado em 1959, soluciona o problema do caminho mais curto num grafo dirigido ou não dirigido com arestas de peso não negativo, em tempo computacional O([m+n]log n) onde m é o número de arestas e n é o número de vértices.

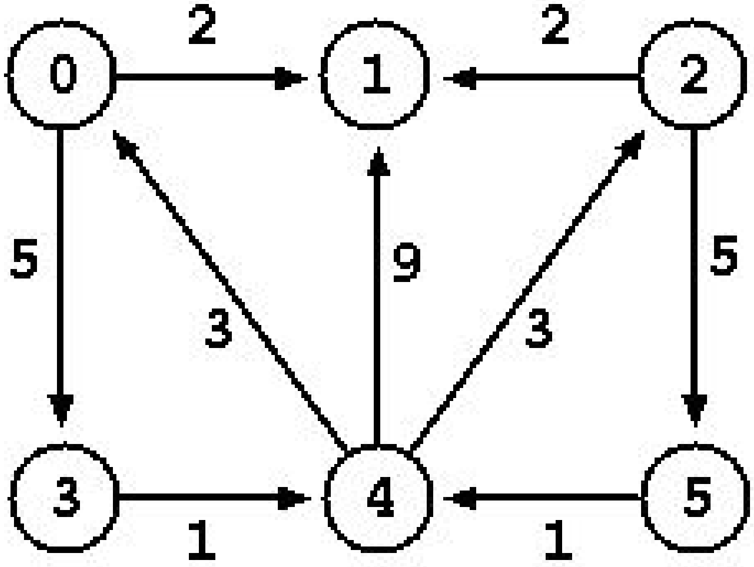


∞
∞
5
b
9
6
6
2
∞
∞
3
11
4
14
9
10
15
0
1
a
∞
7
2

Algoritmo de warshall

Na [ciência da computação](), o **algoritmo de Floyd-Warshall** é um [algoritmo]() que resolve o problema de calcular o [caminho mais curto]() entre todos os pares de [vértices]() em um [grafo orientado]() (com direção) e valorado (com peso). O algoritmo Floyd-Warshall foi publicado por Robert Floyd em 1962. Este algoritmo é o mesmo que foi publicado por [Bernard Roy]() em 1959 e também por [Stephen Warshall]() em 1962 para determinar o fechamento transitivo de um grafo.[1] O formato atual do algoritmo de Floyd-Warshall com três loops de repetição foi descrito por [Peter Ingerman]() em 1962.





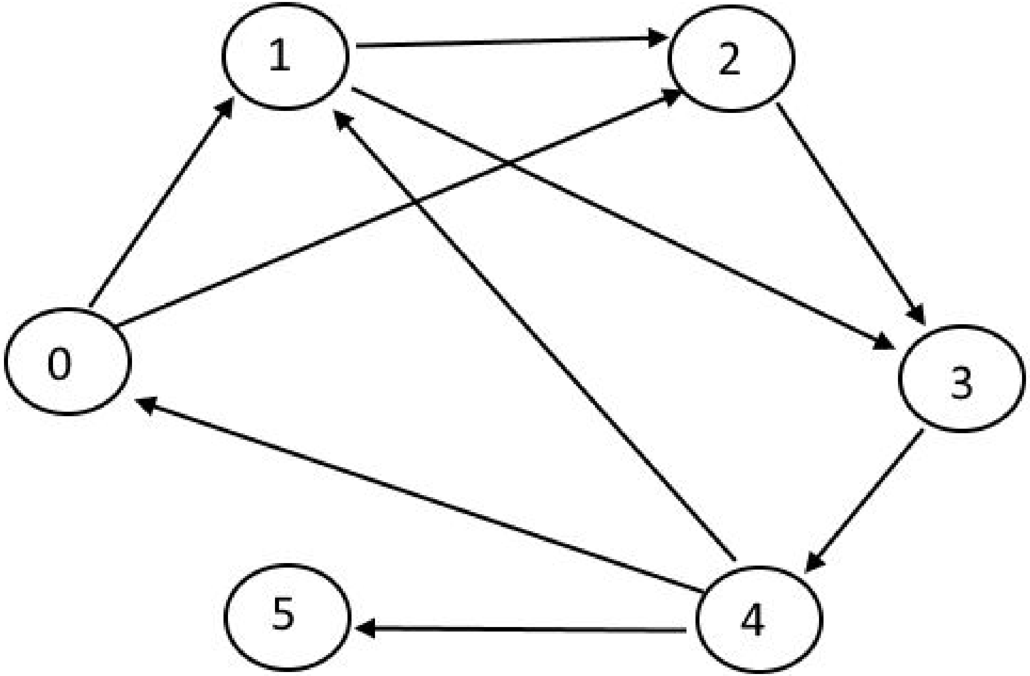
	0	1	2	3	4	5
0	0	2	0	5	0	0
1	0	0	0	0	0	0
2	0	2	0	0	0	5
3	0	0	0	0	1	0
4	3	9	3	0	0	0
5	0	0	0	0	1	0

BFS & DFS

Na teoria dos grafos, **busca em largura** (ou busca em amplitude, também conhecido em inglês por Breadth-First Search - BFS) é um algoritmo de busca em grafos utilizado para realizar uma busca ou travessia num grafo e estrutura de dados do tipo árvore. Intuitivamente, você começa pelo vértice raiz e explora todos os vértices vizinhos. Então, para cada um desses vértices mais próximos, exploramos os seus vértices vizinhos inexplorados e assim por diante, até que ele encontre o alvo da busca.

Na teoria dos grafos, **busca em profundidade** (ou busca em profundidade-primeiro, também conhecido em inglês por Depth-First Search - DFS) é um algoritmo usado para realizar uma busca ou travessia numa árvore, estrutura de árvore ou grafo. Intuitivamente, o algoritmo começa num nó raiz (selecionando algum nó como sendo o raiz, no caso de um grafo) e explora tanto quanto possível cada um dos seus ramos, antes de retroceder(backtracking).

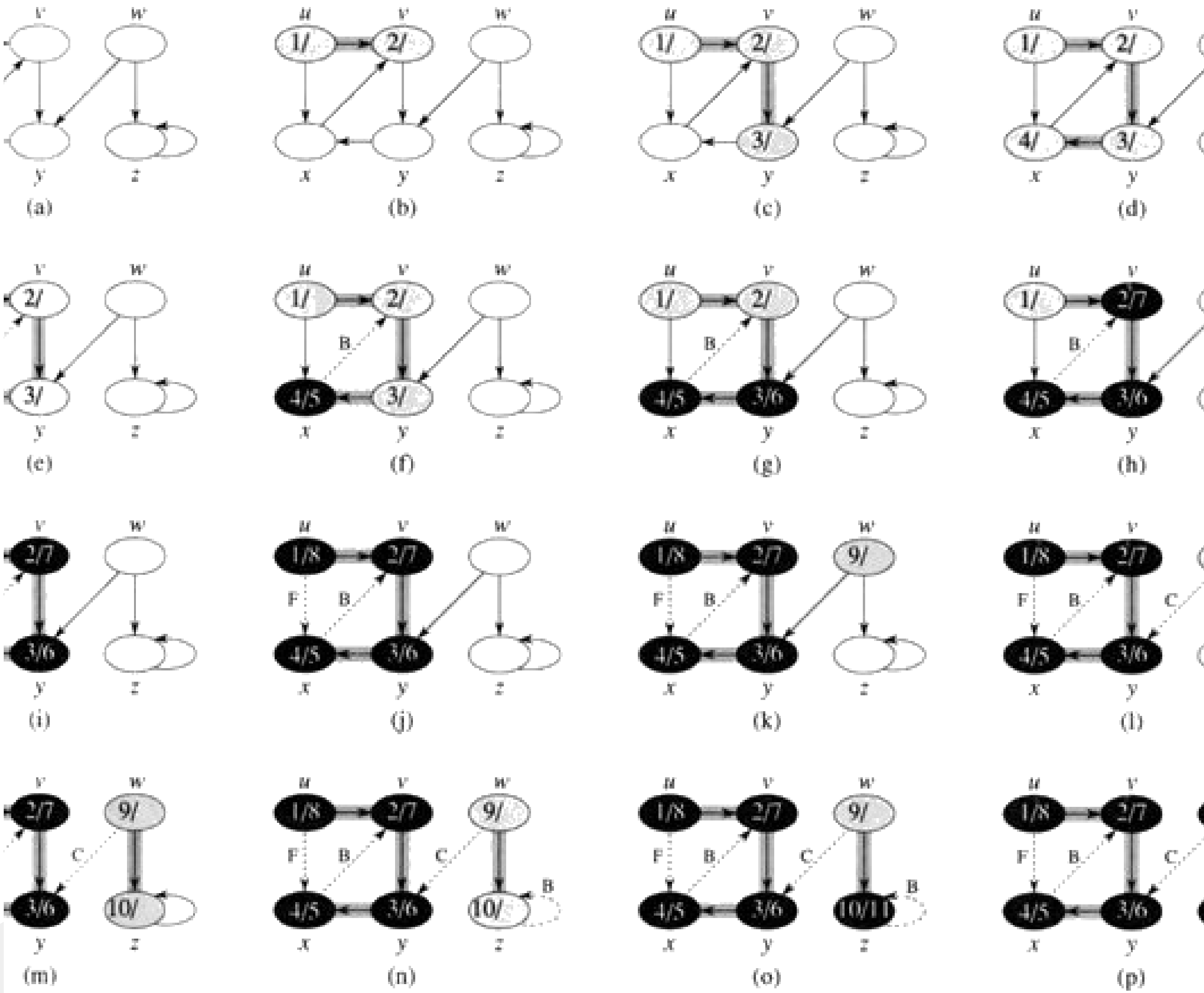
BFS



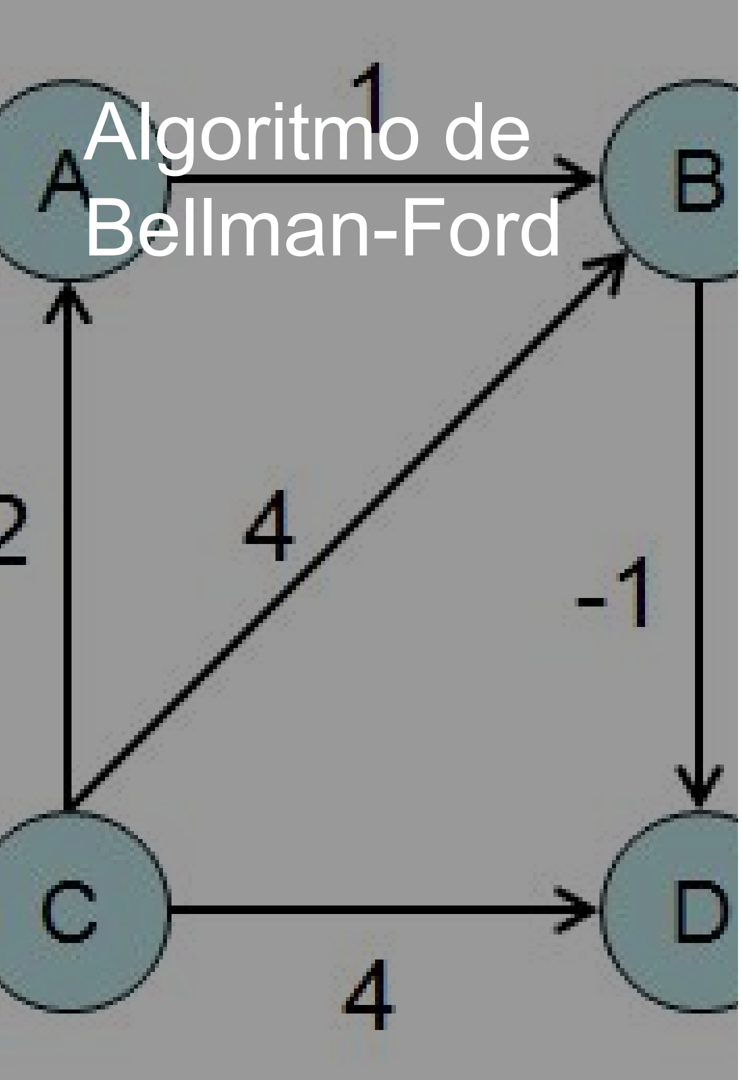
BFS starting from Node 0

0 2 1 3 4 5

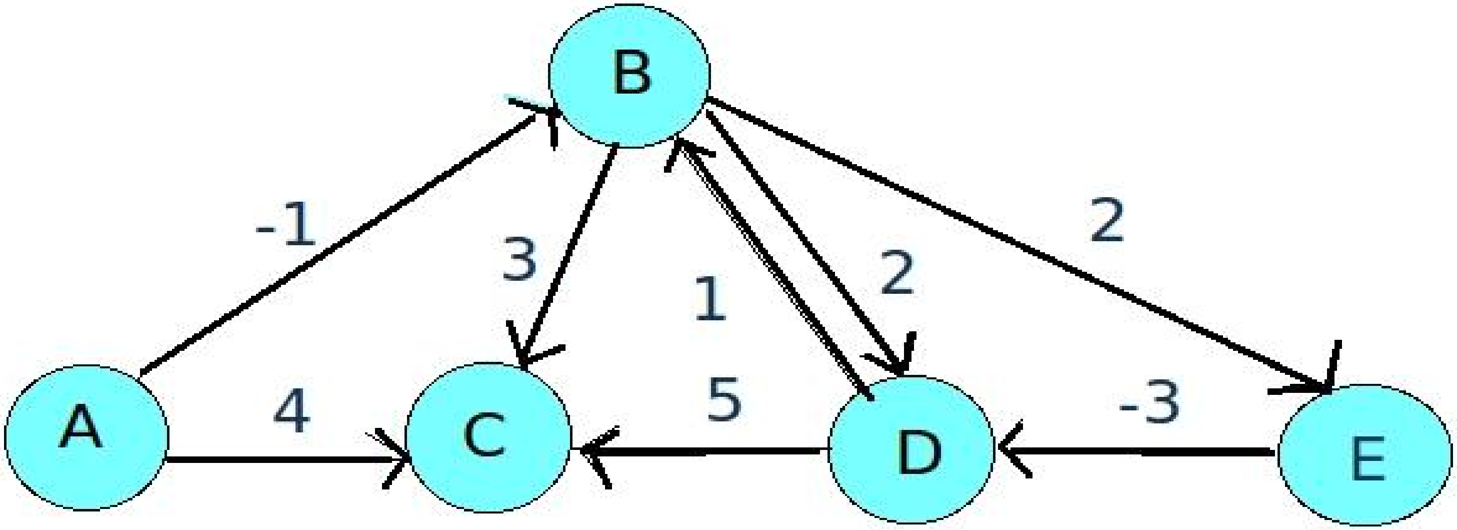
DFS



Algoritmo de Bellman-Ford



O **Algoritmo de Bellman-Ford** é um algoritmo de busca de caminho mínimo em um dígrafo ponderado, ou seja, cujas arestas têm peso, inclusive negativo. O Algoritmo de Dijkstra resolve o mesmo problema, num tempo menor, porém exige que todas as arestas tenham pesos positivos. Portanto, o algoritmo de Bellman-Ford é normalmente usado apenas quando existem arestas de peso negativo.

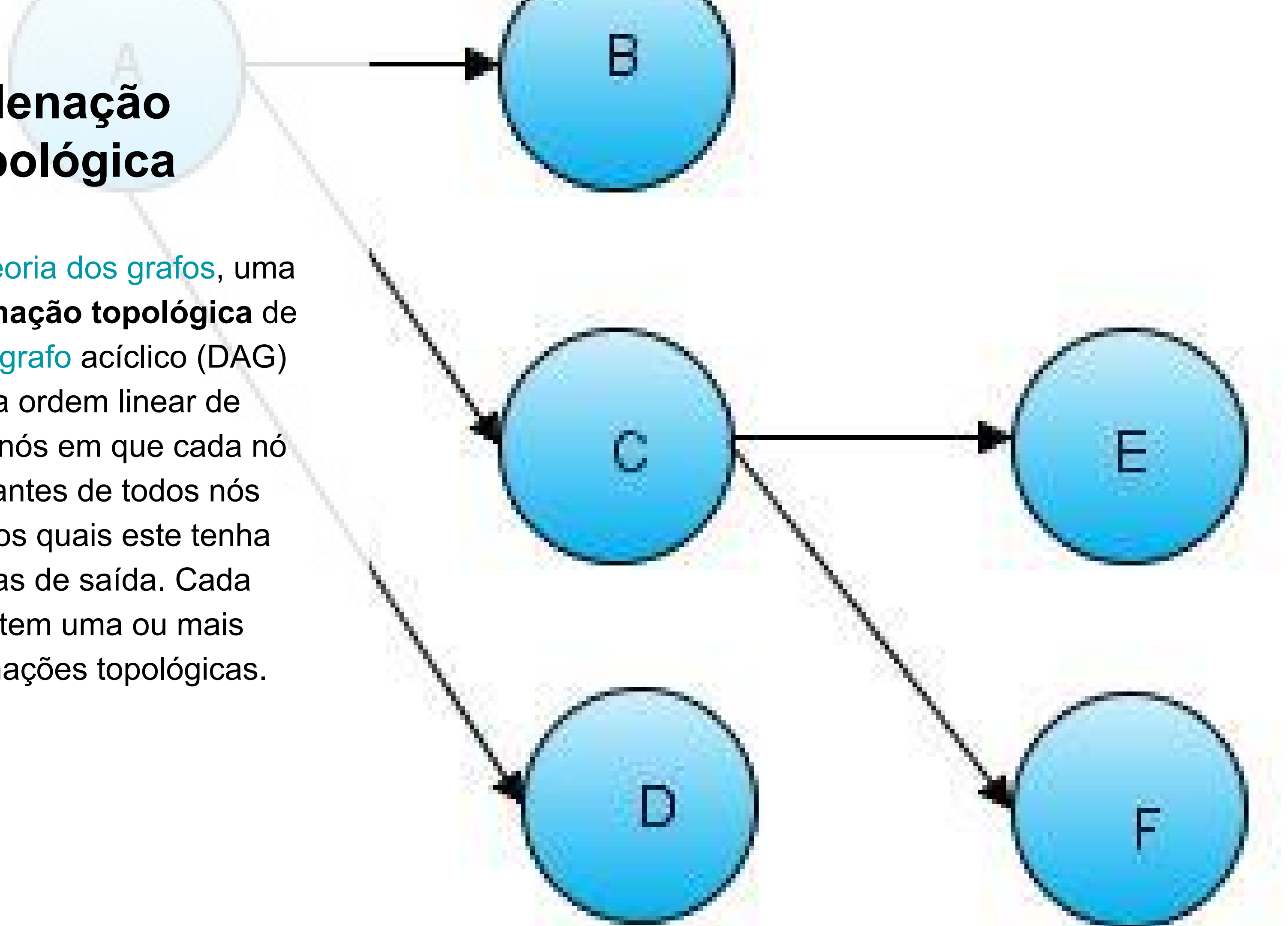


Shortest distance from source node A, to all the other nodes.

	A	B	C	D	E
1st iteration	0	inf	inf	inf	inf
2nd iteration	0	-1	4	inf	inf
3rd iteration	0	-1	2	1	1
4th iteration	0	-1	2	-2	1

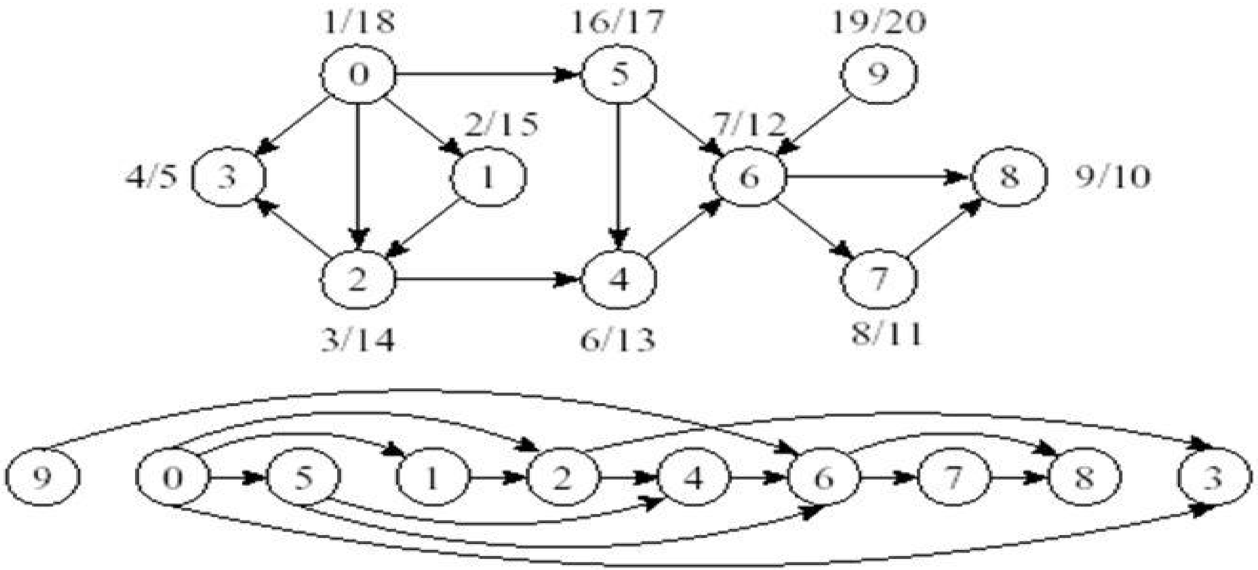
Ordenação Topológica

Em teoria dos grafos, uma **ordenação topológica** de um digrafo acíclico (DAG) é uma ordem linear de seus nós em que cada nó vem antes de todos nós para os quais este tenha arestas de saída. Cada DAG tem uma ou mais ordenações topológicas.



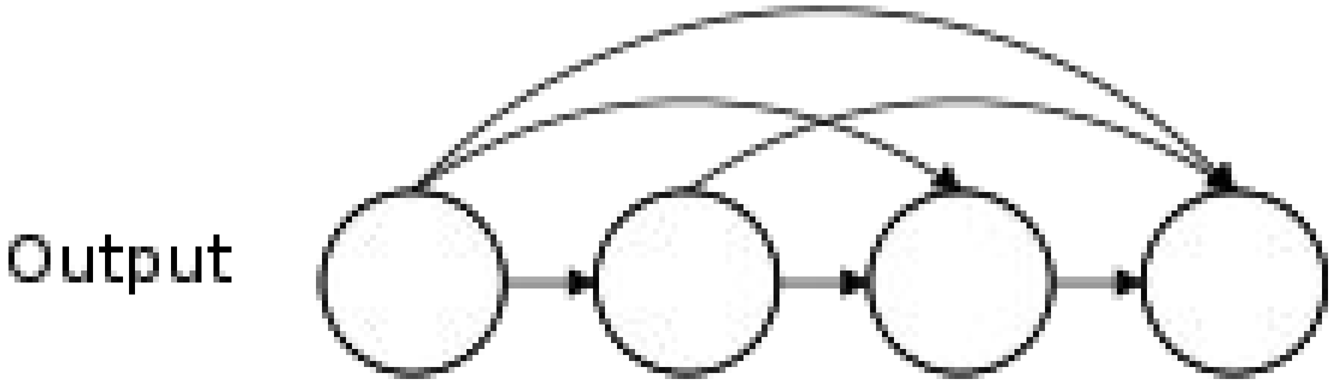
Ordenação Topológica

- Uma aresta direcionada (u,v) indica que a atividade u tem que ocorrer antes da atividade v



Fecho Transitivo

Se a própria relação binária é transitiva, então o fecho transitivo é a própria relação; senão, o fecho transitivo é uma outra relação. Por exemplo, se X é um conjunto de aeroportos e $x\ R\ y$ significa "existe um voo direto do aeroporto x para o aeroporto y", então o fecho transitivo de R sobre X é a relação R^+: "é possível voar a partir de x para y em um ou mais voos."


Input
Output