



INSTITUTO FEDERAL DE  
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA  
SUL-RIO-GRANDENSE  
Campus Passo Fundo



# **ESTRUTURA DE DADOS II**

**Prof. Adilso Nunes de Souza**



# TIPOS DE GRAFOS

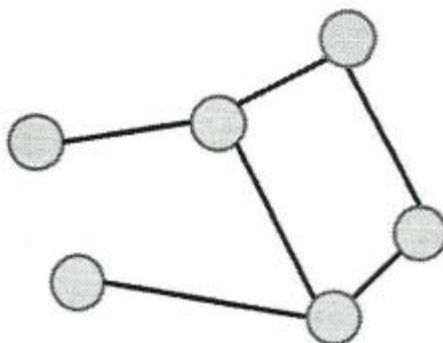
- Grafo trivial: é a forma mais simples de grafo que existe. Trata-se de um grafo que possui um único vértice e nenhuma aresta ou laço.





# TIPOS DE GRAFOS

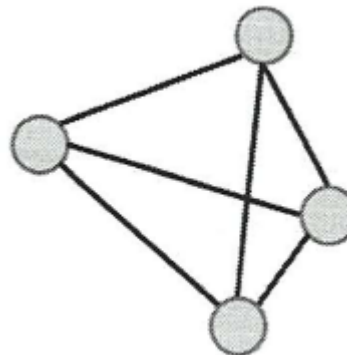
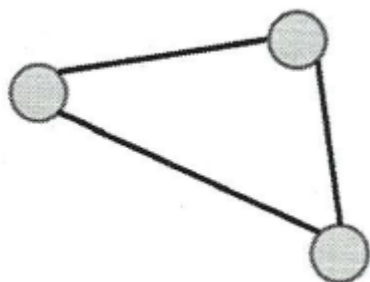
- Grafo Simples: trata-se de um grafo não direcionado, sem laços e sem arestas paralelas





# TIPOS DE GRAFOS

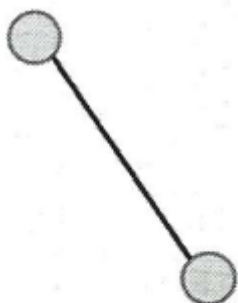
- Grafo Completo: consiste em um grafo simples (não direcionado, sem laços e sem arestas paralelas), onde cada vértice se conecta a todos os outros vértices do grafo.



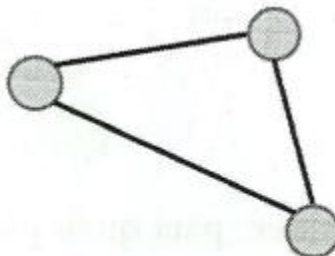


# TIPOS DE GRAFOS

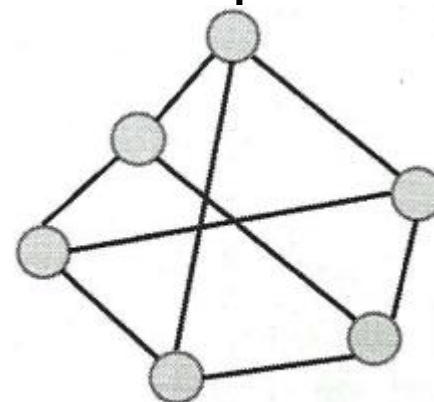
- Grafo Regular: é todo o grafo cujos vértices possuem o mesmo grau (número de arestas ligadas a ele).
- Todo o grafo completo é também regular, mas nem todo o grafo regular é dito completo.



Grau = 1



Grau = 2

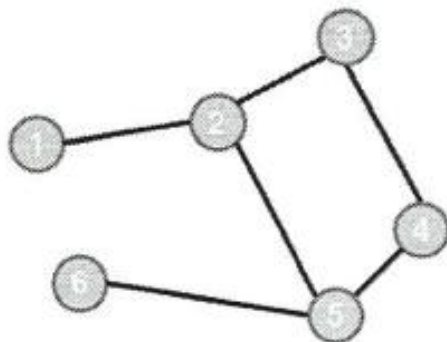


Grau = 3

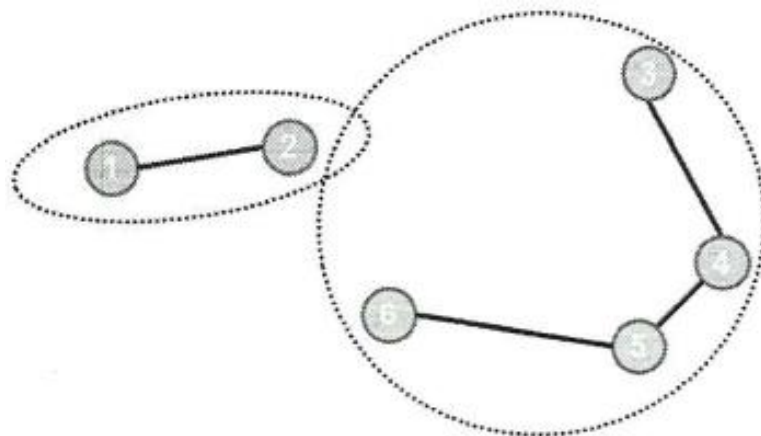


# TIPOS DE GRAFOS

- Subgrafo: É um grafo construído de tal forma que seus vértices e suas arestas estão contidos em outro grafo.



GRAFO

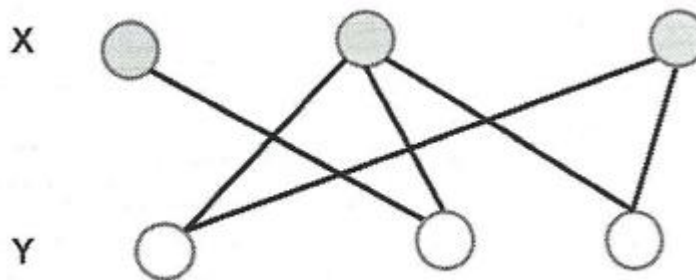


SUBGRAFOS



# TIPOS DE GRAFOS

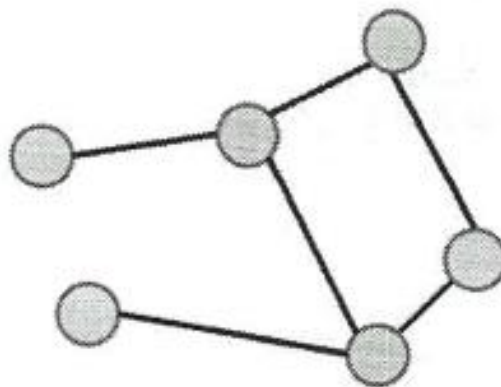
- Grafo Bipartido: é o grafo cujo conjunto de vértices pode ser dividido em dois subconjuntos sem intersecção. Já as arestas conectam apenas os vértices que estão em subconjuntos diferentes, ou seja uma aresta nunca conecta vértices do mesmo subconjunto.





# TIPOS DE GRAFOS

- Grafo Conexo: é todo o grafo que, para quaisquer dois vértices distintos, sempre existe um caminho que os une.



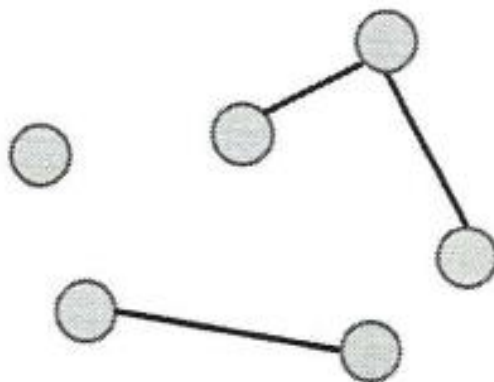
GRAFO CONEXO





# TIPOS DE GRAFOS

- Grafo Desconexo: contém no mínimo duas partes, é um grafo onde existe pelo menos um vértice que não pode ser conectado a outro por um caminho.



GRAFO DESCONEXO



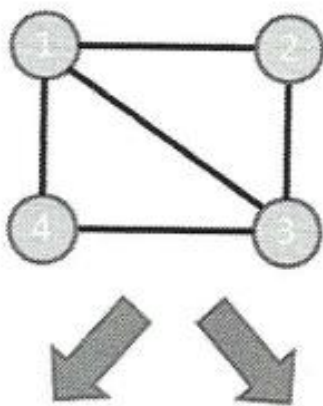
# TIPOS DE GRAFOS

- Grafos Isomorfos: são ditos isomorfos se existir uma função que faça o mapeamento de vértices e arestas de modo que os dois grafos se tornem coincidentes.
- Condições mínimas para que dois grafos sejam isomorfos:
  - Possuírem o mesmo número de vértices
  - Possuírem o mesmo número de arestas
  - Possuírem o mesmo número de vértices de grau  $n$ , para qualquer valor  $n$  entre 0 e o número de vértices que o grafo contém.



# TIPOS DE GRAFOS

- Grafo Isomorfos:



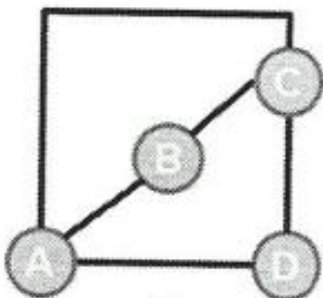
Grau

$f(1) = A$

$f(2) = B$

$f(3) = C$

$f(4) = D$



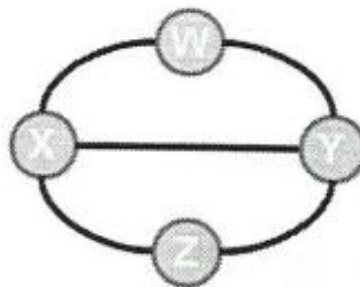
Grau

$f(1) = X$

$f(2) = W$

$f(3) = Y$

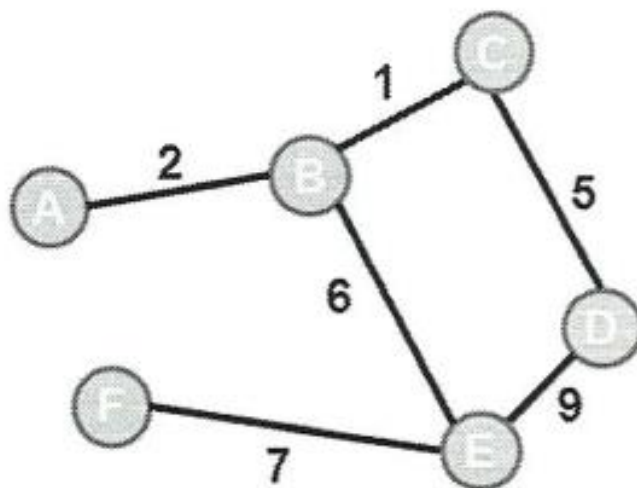
$f(4) = Z$





# TIPOS DE GRAFOS

- Grafo Ponderado: ocorre quando as arestas possuem um peso (valor numérico) associado a cada uma delas.





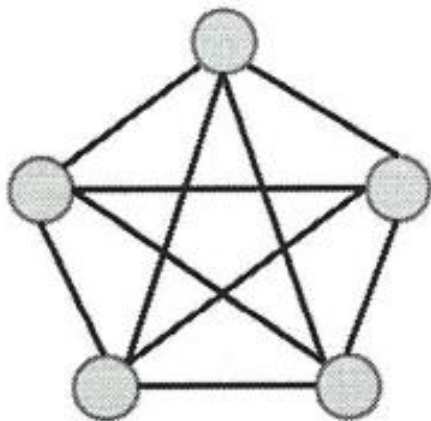
# TIPOS DE GRAFOS

- Grafo Hamiltoniano: é um tipo especial de grafo que possui um caminho que visita todos os seus vértices apenas uma vez, a este caminho denomina-se caminho hamiltoniano.
- Um ciclo hamiltoniado é um ciclo no qual cada vértice é visitado exatamente uma vez, retornando ao seu ponto de partida (esse é o único que se repete)

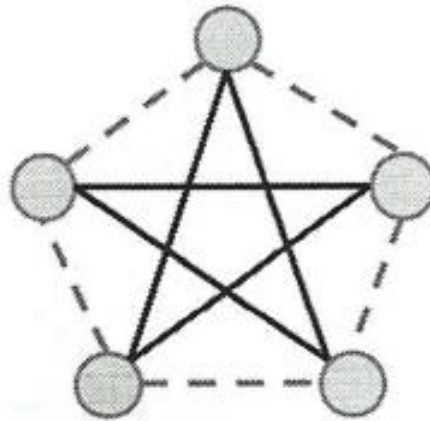


# TIPOS DE GRAFOS

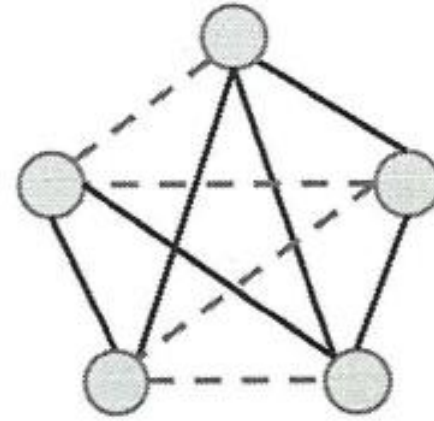
- Grafo Hamiltoniano:



GRAFO HAMILTONIANO



CICLO HAMILTONIANO

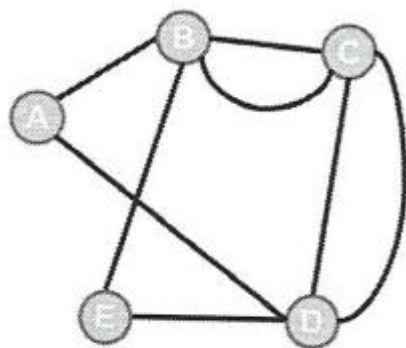


CAMINHO  
HAMILTONIANO

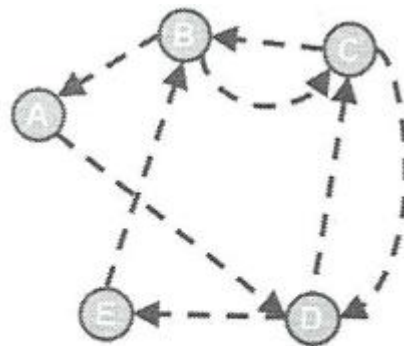


# TIPOS DE GRAFOS

- Grafo Euleriano: é um tipo de grafo que possui um ciclo que visita todas as suas arestas apenas uma vez, iniciando e terminando no mesmo vértice.
- Este ciclo dá-se o nome de ciclo euleriano



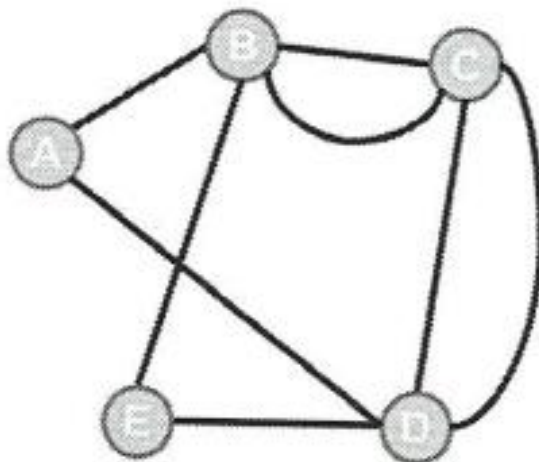
GRAFO EULERIANO



CICLO EULERIANO  
C-D-C-B-A-D-E-B-C

# TIPOS DE GRAFOS

- Teorema de Euler: Um grafo conexo (não necessariamente simples)  $G$  é euleriano se e somente se todos os seus vértices tem grau par.



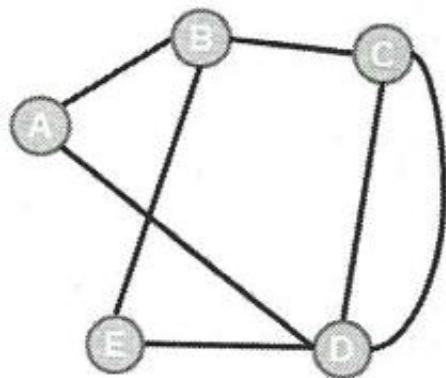
GRAFO EULERIANO



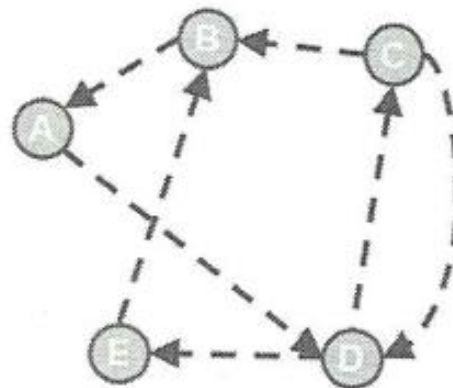


# TIPOS DE GRAFOS

- Um grafo conexo (não necessariamente simples)  $G$  é **semieuleriano** se, e somente se, no máximo, dois vértices têm grau ímpar.



GRAFO  
SEMIEULERIANO

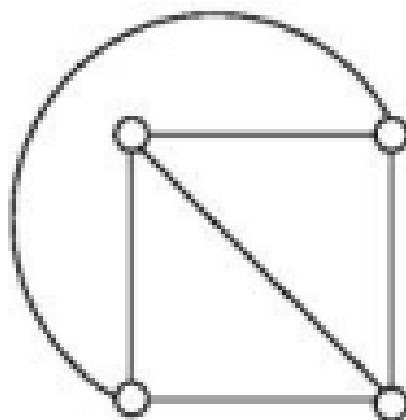


CAMINHO EULERIANO  
C-D-C-B-A-D-E-B



# TIPOS DE GRAFOS

- Grafo Planar: É um grafo representado graficamente no plano de tal forma que não haja cruzamento de suas arestas.





# ALGORITMOS DE BUSCA

- A operação de busca consiste em explorar o grafo de uma maneira específica.
- Trata-se de um processo sistemático para caminhar por seus vértices e arestas.
- As operações de busca são utilizadas para resolver uma série de problemas em grafos, para determinados problemas, a busca necessita visitar todos os vértices, já para outros necessita visitar apenas um subconjunto dos vértices.



# TIPOS DE BUSCA

- Existem vários tipos de busca que podemos realizar em um grafo, as três principais são:
  - Busca em profundidade
  - Busca em largura
  - Busca pelo menor caminho



# BUSCA EM PROFUNDIDADE

- O algoritmo de busca em profundidade é descrito por BACKES, 2016, da seguinte forma:
  - Partindo de um vértice inicial, a busca explora o máximo possível cada um dos vizinhos de um vértice antes de retroceder (backtracking).
  - Inicia em um vértice e se aprofunda nos vértices vizinhos até encontrar o alvo da busca ou um vértice sem vizinho que possa ser visitado.



# BUSCA EM PROFUNDIDADE

- A busca em profundidade, do inglês (depth-first search) DFS, é um algoritmo para caminhar no grafo.
- A estratégia é buscar o mais profundo no grafo sempre que possível.
- As arestas são exploradas a partir do vértice  $v$  mais recentemente descoberto que ainda possui arestas não exploradas saindo dele.



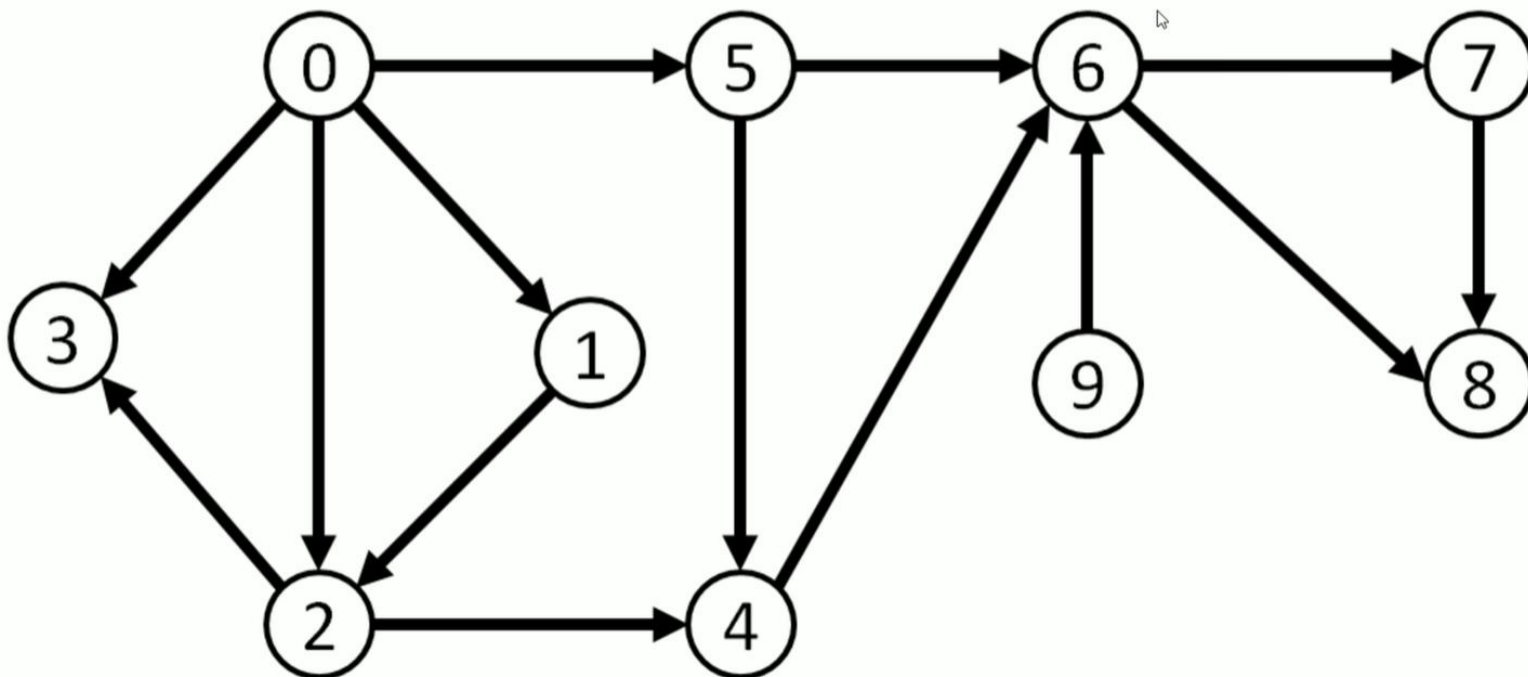
# BUSCA EM PROFUNDIDADE

- Quando todas as arestas adjacentes a  $v$  tiverem sido exploradas a busca anda para trás para explorar vértices que saem do vértice do qual  $v$  foi descoberto.
- O algoritmo é a base para muitos outros algoritmos importantes, tais como verificação de grafos acíclicos, ordenação topológica e componentes fortemente conectados.



# ORDENAÇÃO TOPOLÓGICA

- Busca obter os termos finais para cada vértice do grafo usando a DFS.

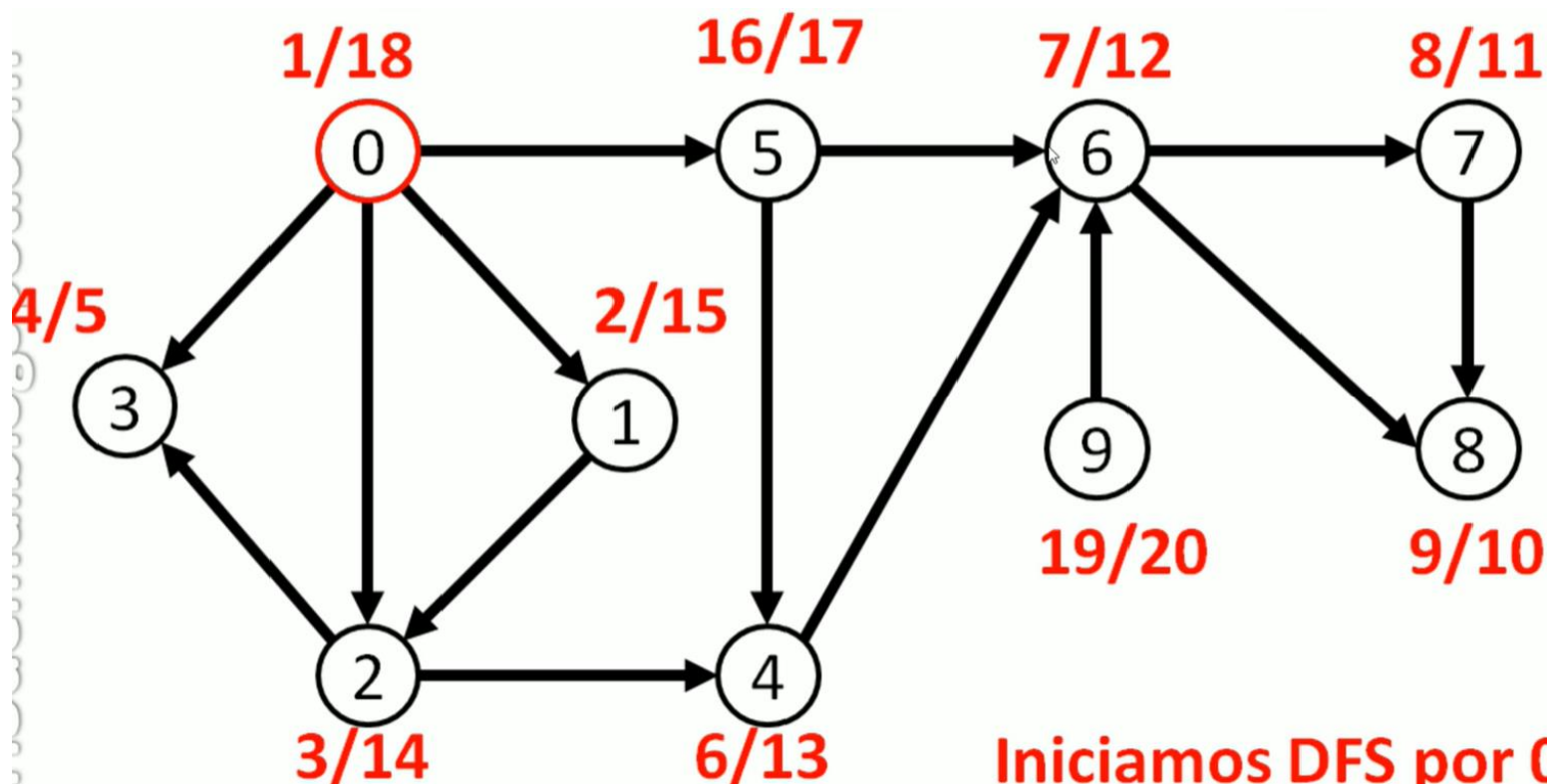






# ORDENAÇÃO TOPOLÓGICA

- 9 - 0 - 5 - 1 - 2 - 4 - 6 - 7 - 8 - 3



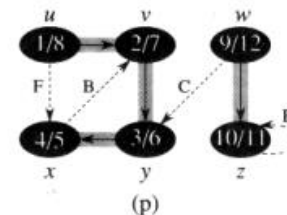
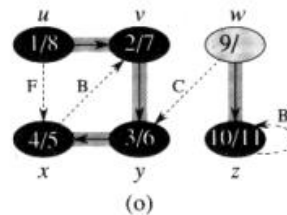
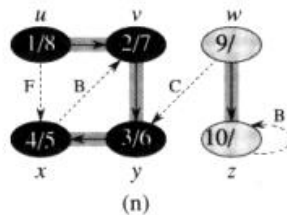
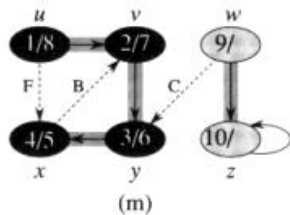
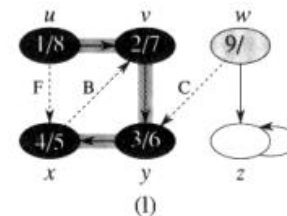
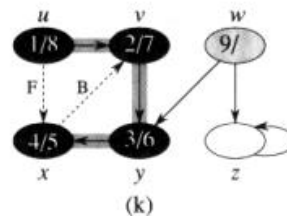
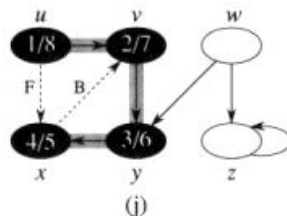
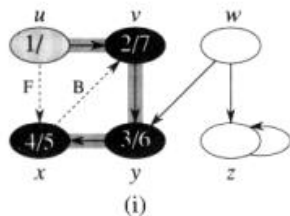
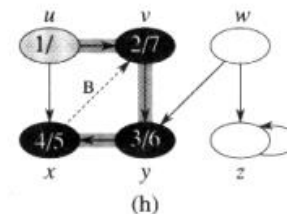
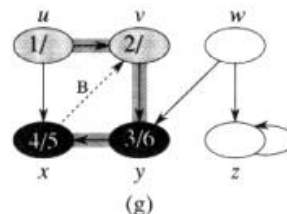
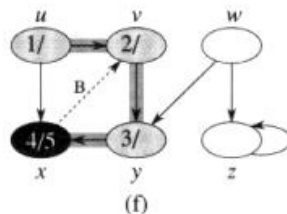
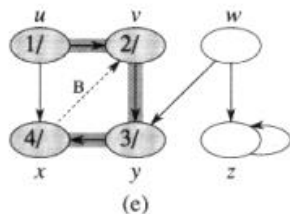
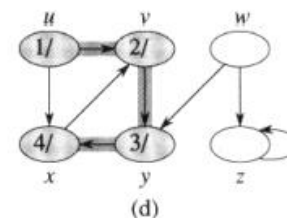
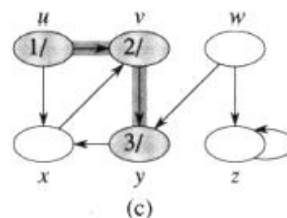
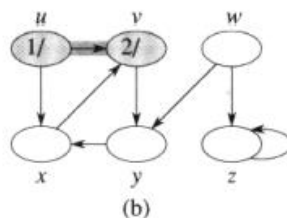
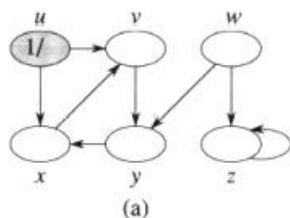


# BUSCA EM PROFUNDIDADE

- Para acompanhar o progresso do algoritmo diferentes autores indicam cores para cada vértice: branco (white), cinza (gray) ou preto (black).
  - white: não visitados ainda.
  - gray: vértice descoberto mas que não teve a sua lista de adjacência totalmente examinada.
  - black: vértice descoberto que já teve a sua lista de adjacência totalmente examinada e está terminado.

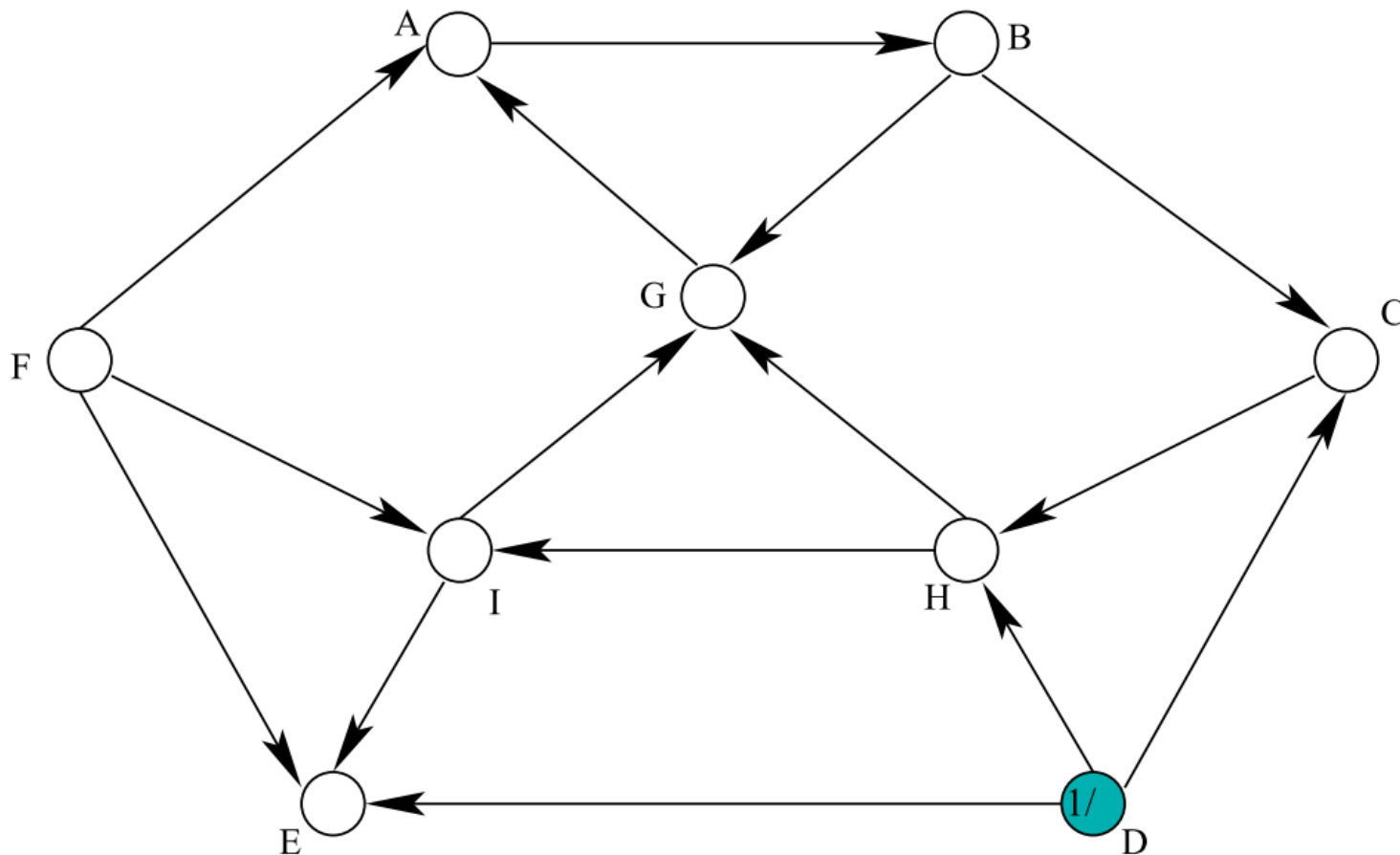


# EXECUÇÃO DO ALGORITMO



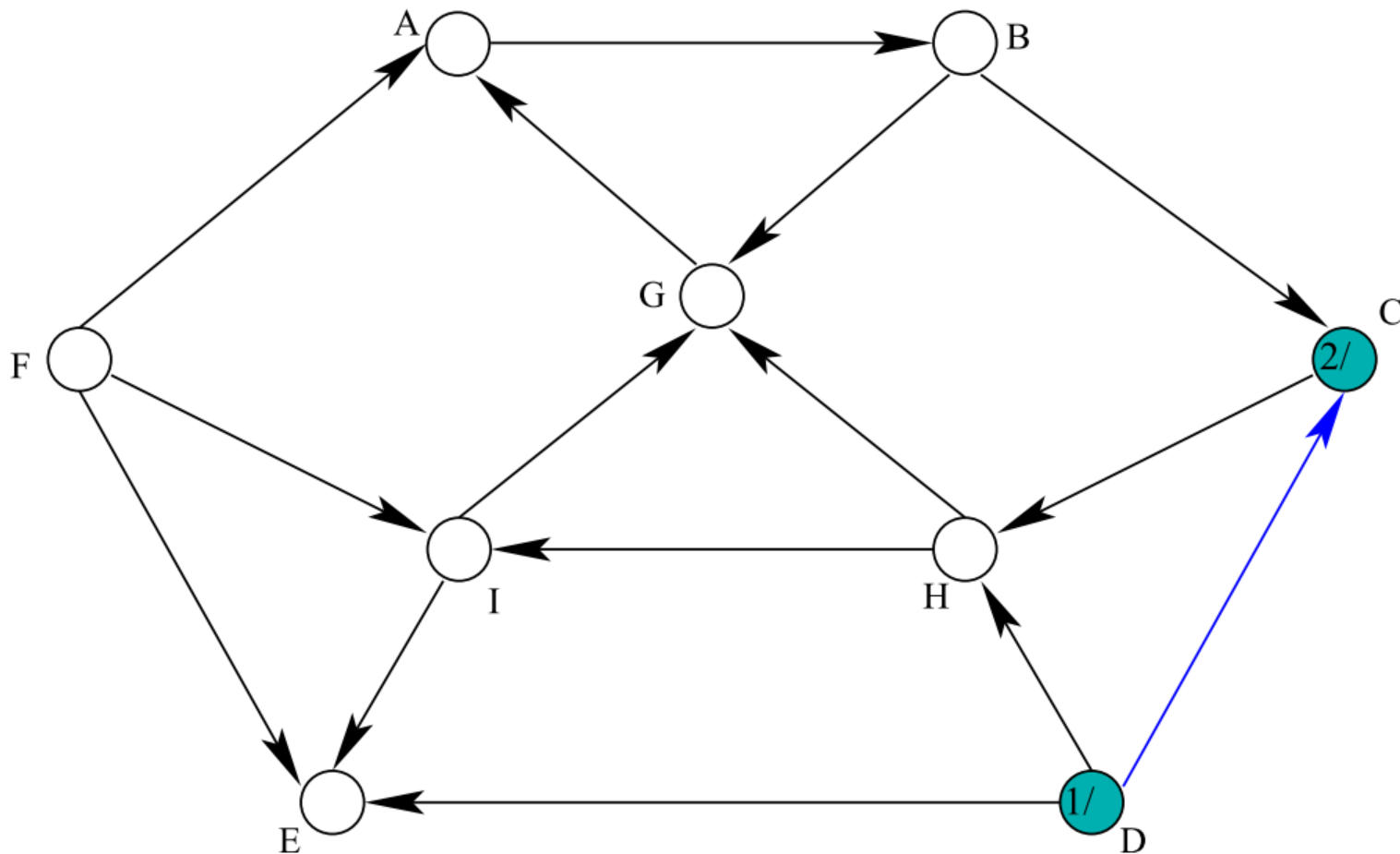


# EXEMPLO DE EXECUÇÃO



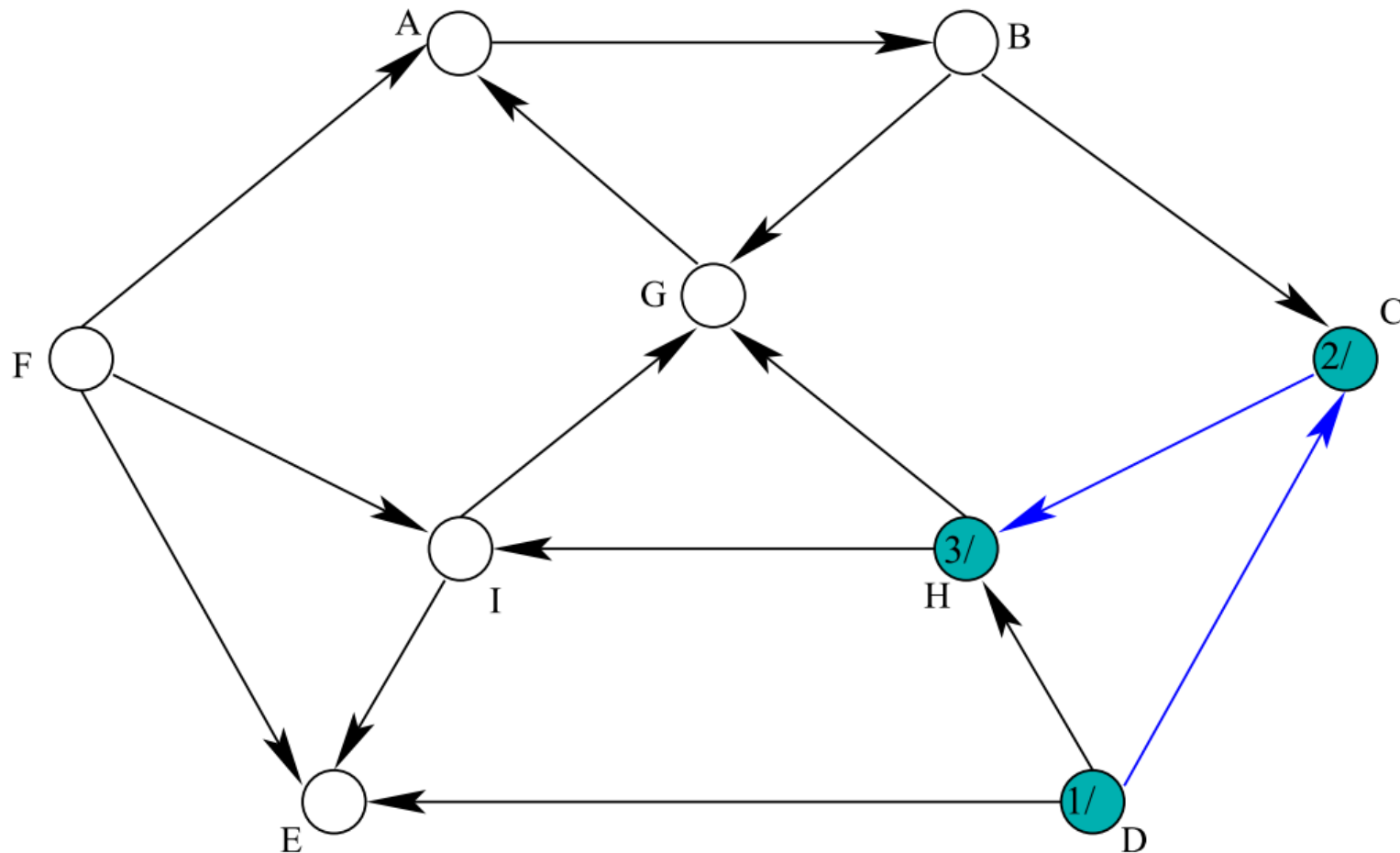


# EXEMPLO DE EXECUÇÃO



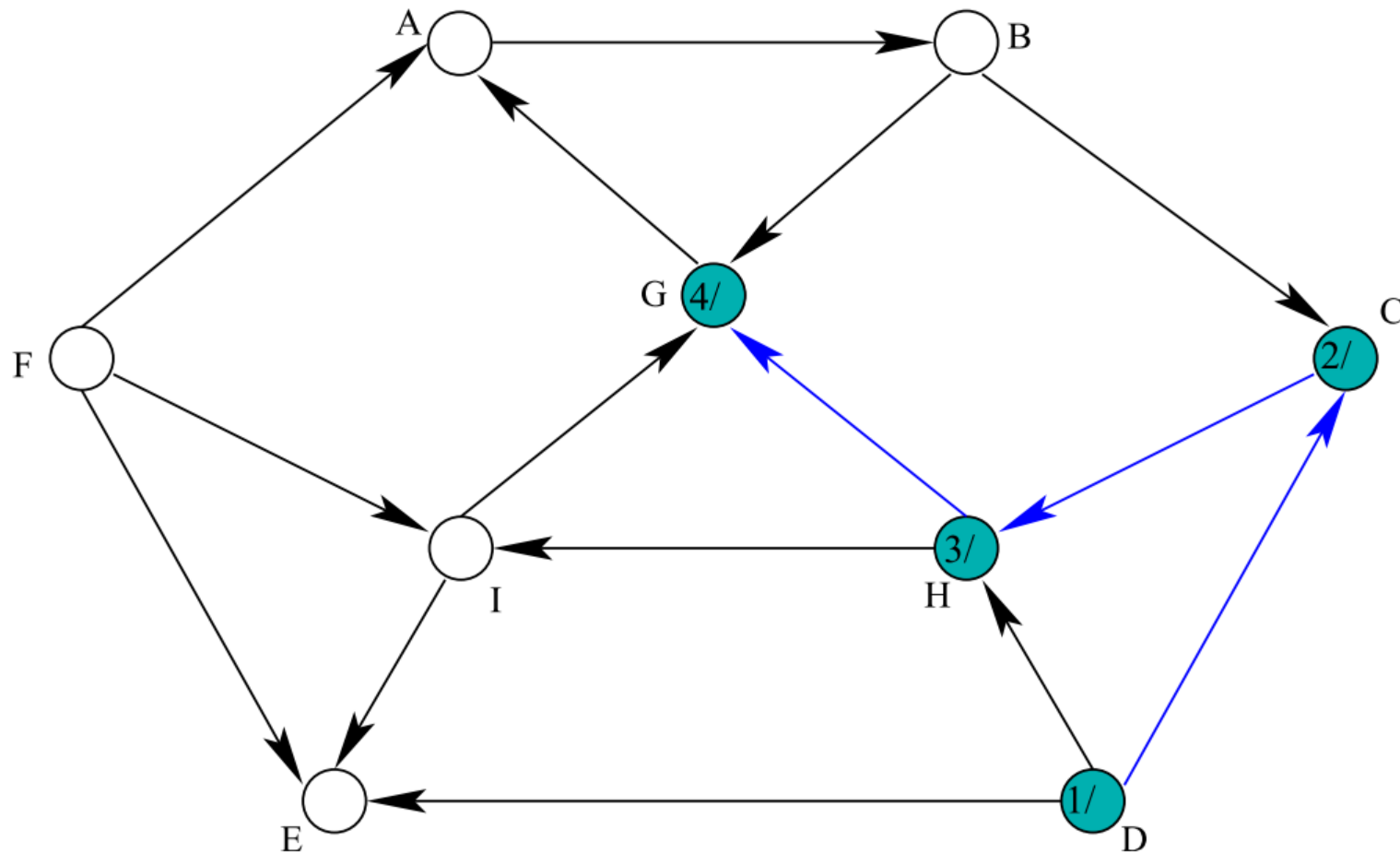


# EXEMPLO DE EXECUÇÃO



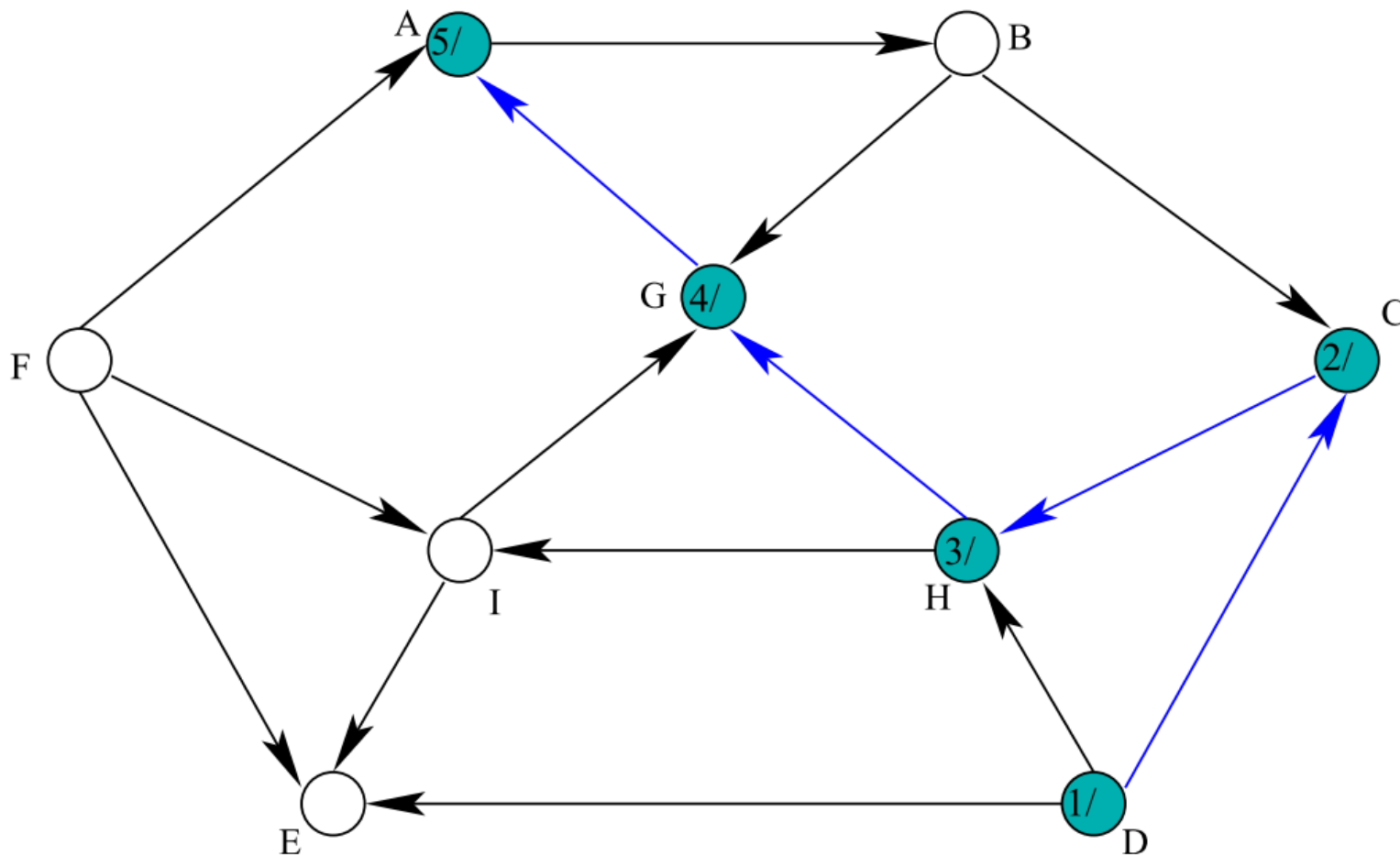


# EXEMPLO DE EXECUÇÃO





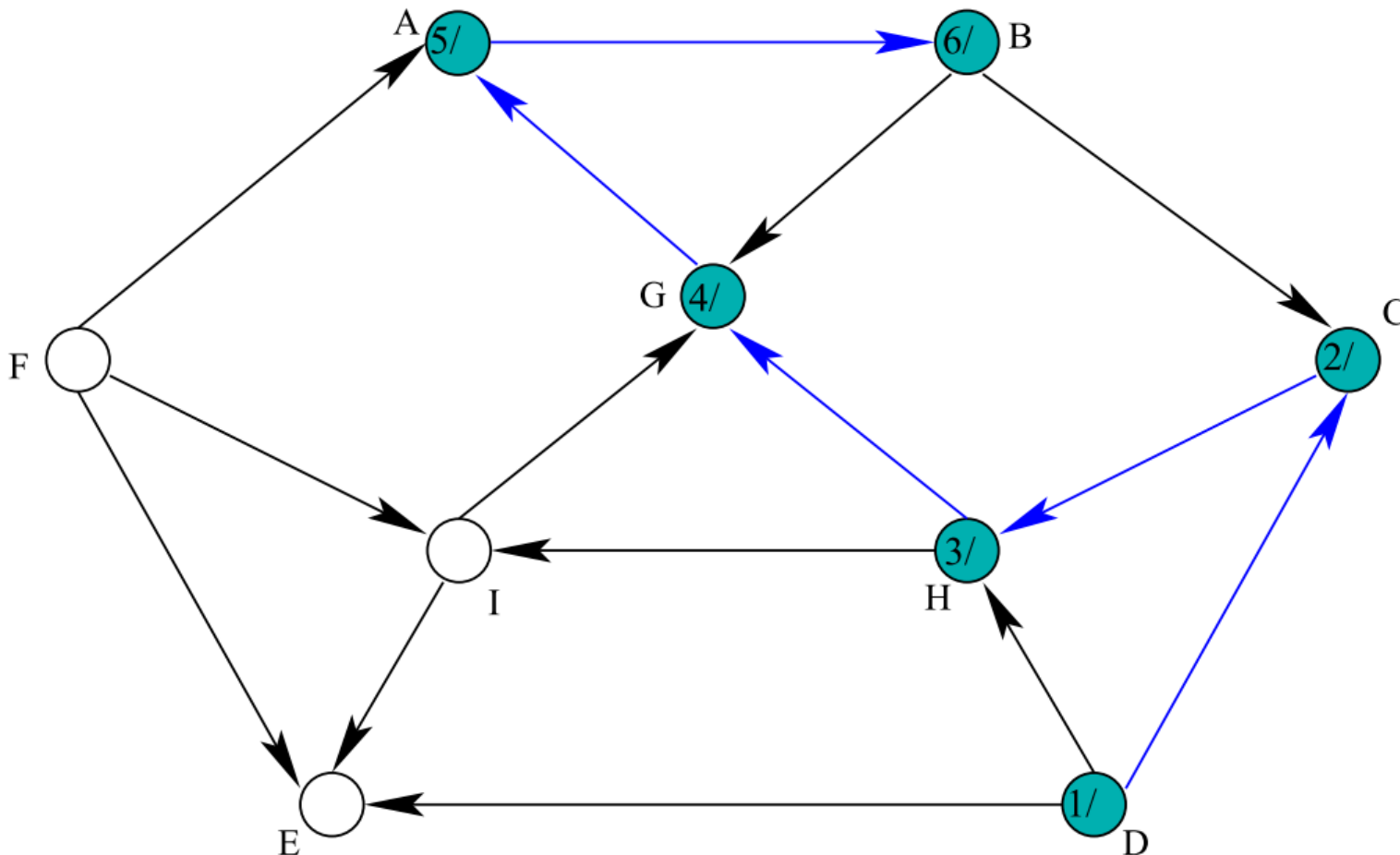
# EXEMPLO DE EXECUÇÃO





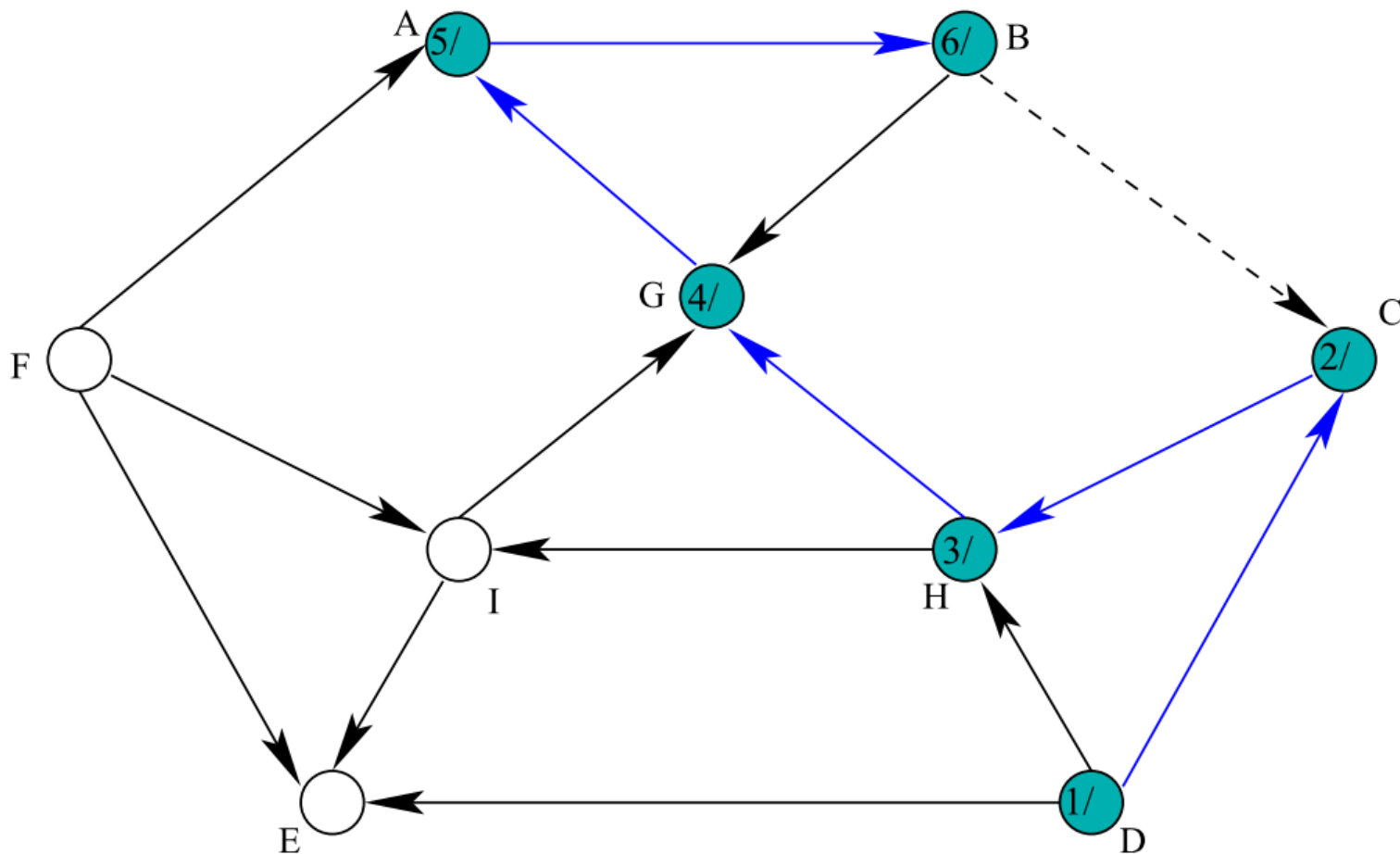


# EXEMPLO DE EXECUÇÃO



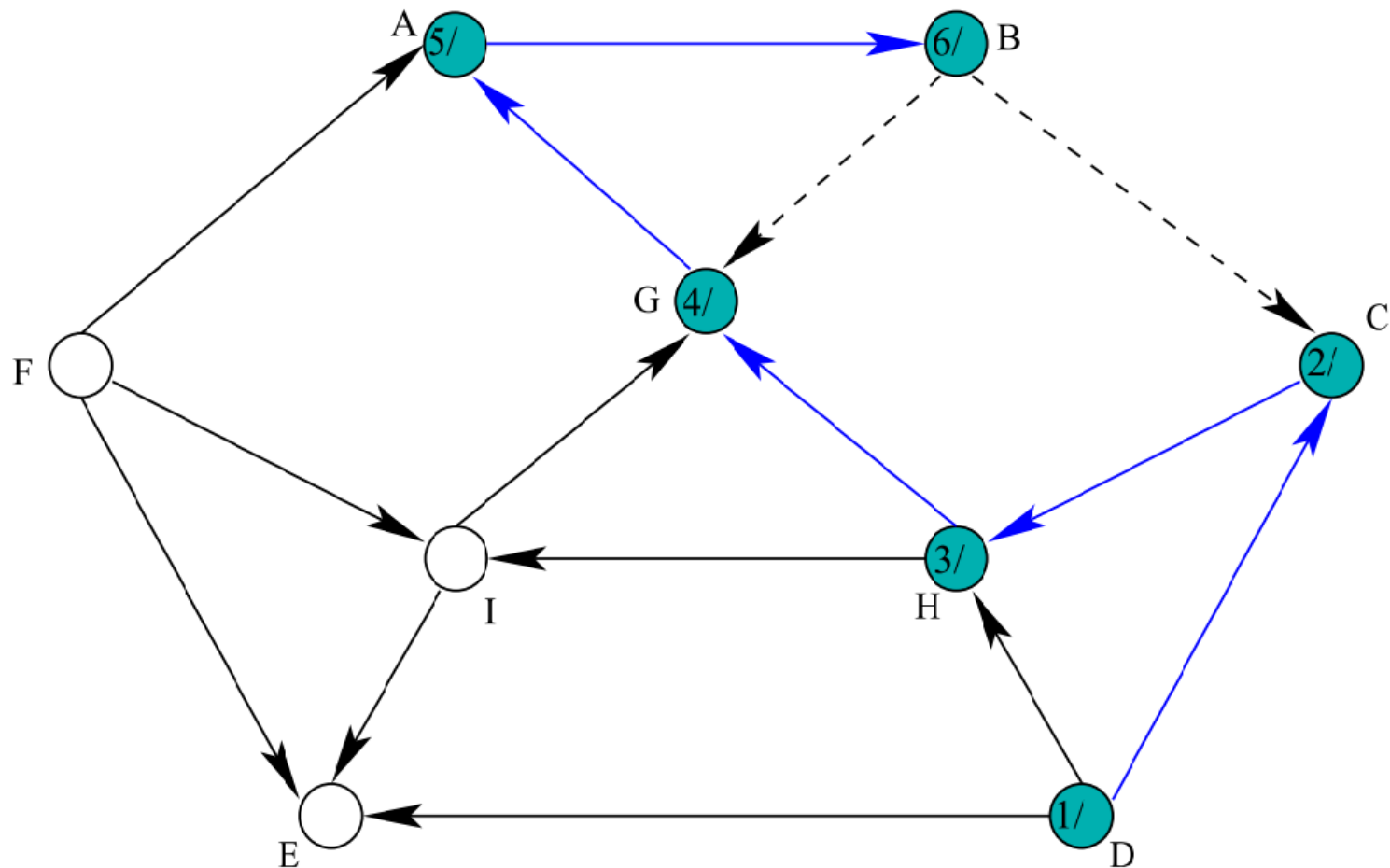


# EXEMPLO DE EXECUÇÃO



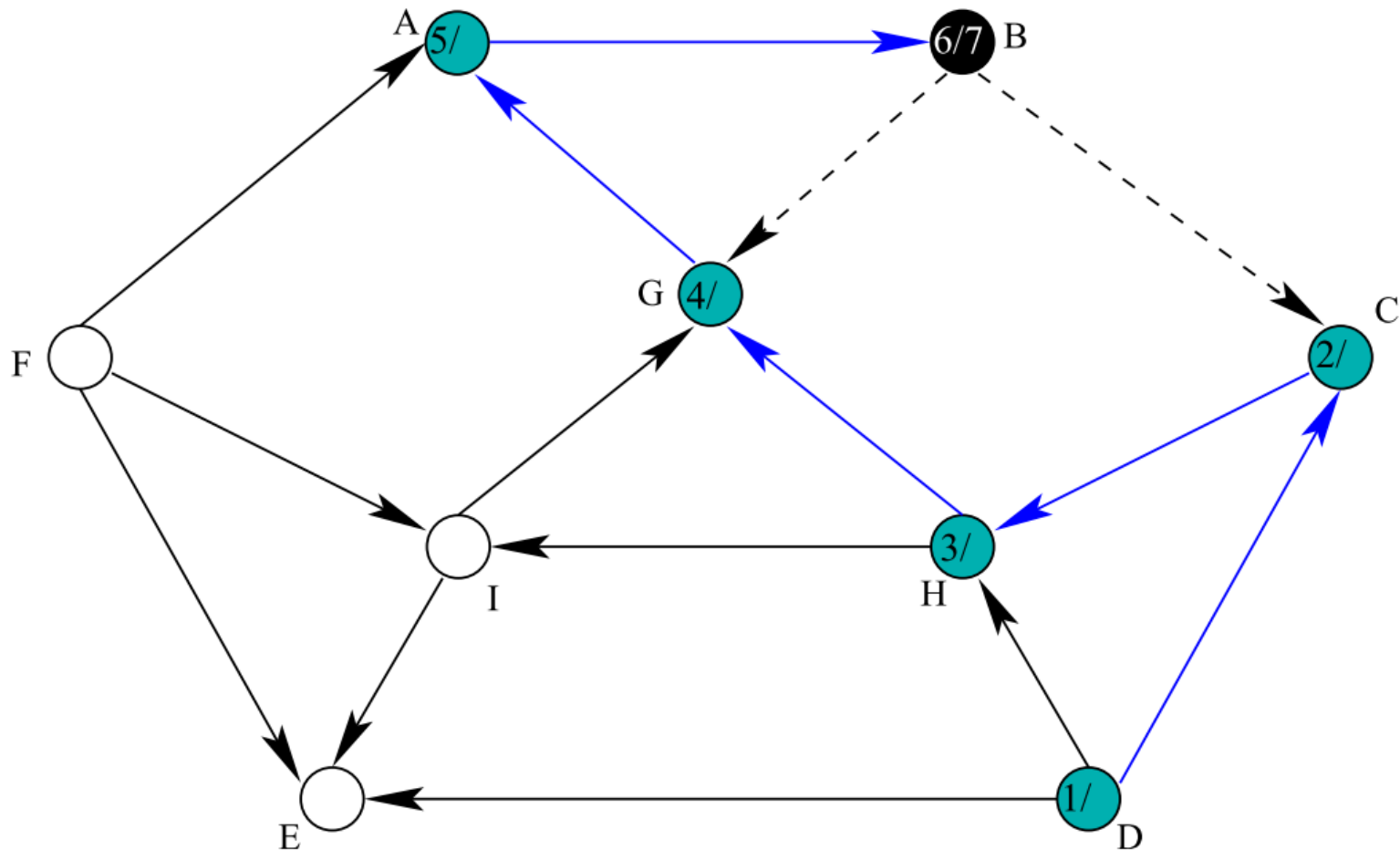


# EXEMPLO DE EXECUÇÃO



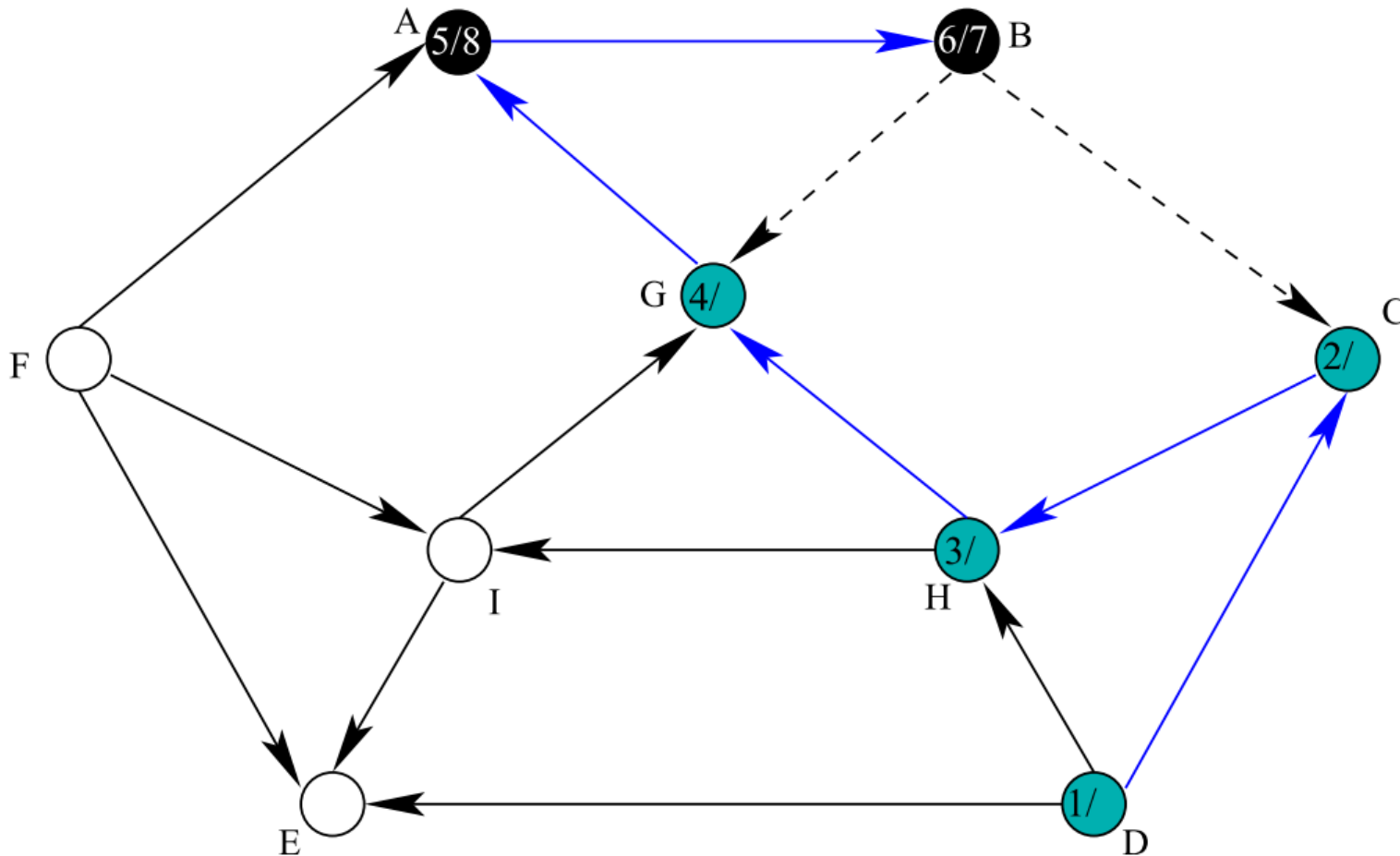


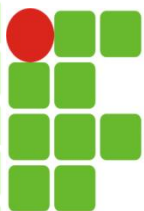
# EXEMPLO DE EXECUÇÃO



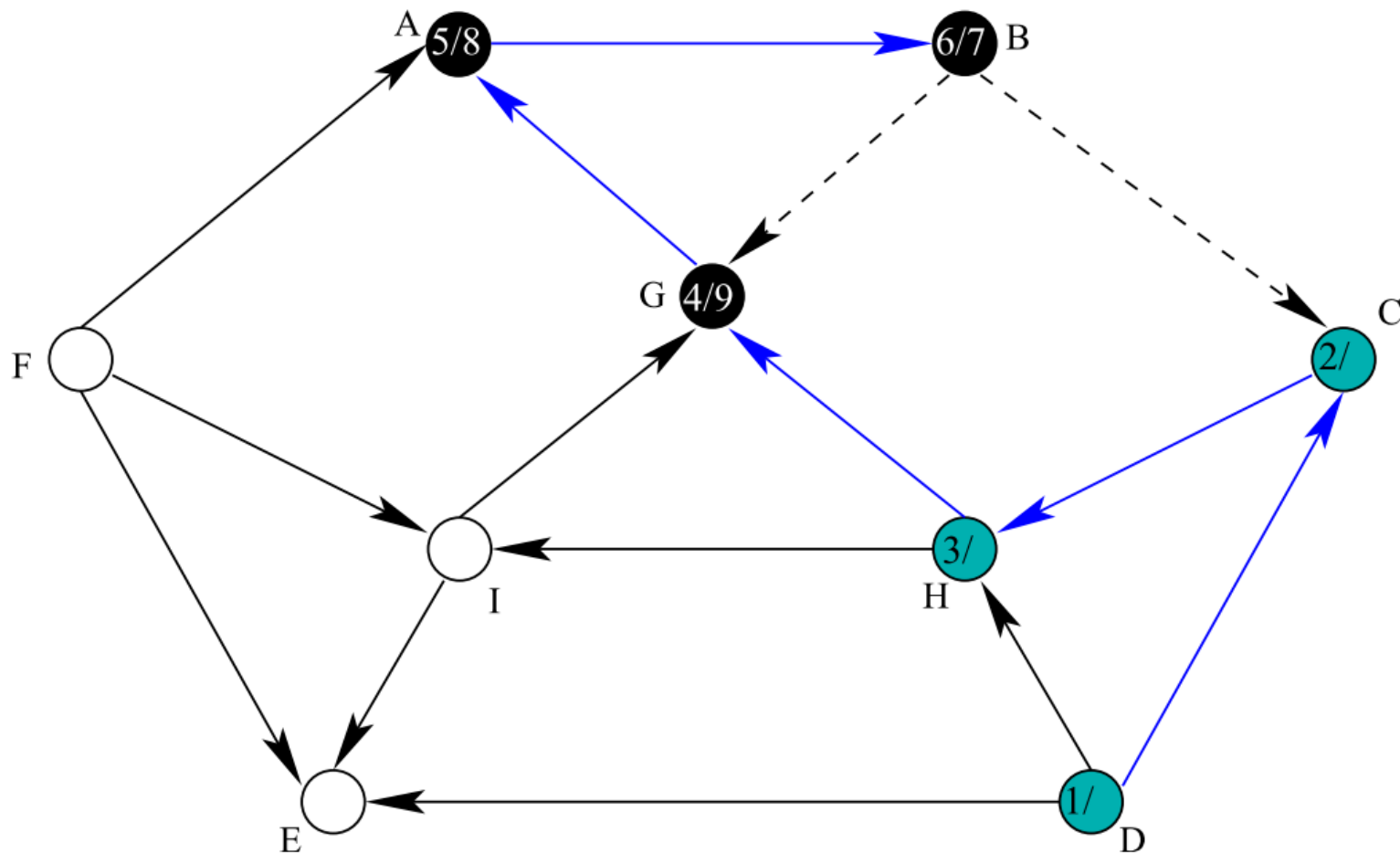


# EXEMPLO DE EXECUÇÃO



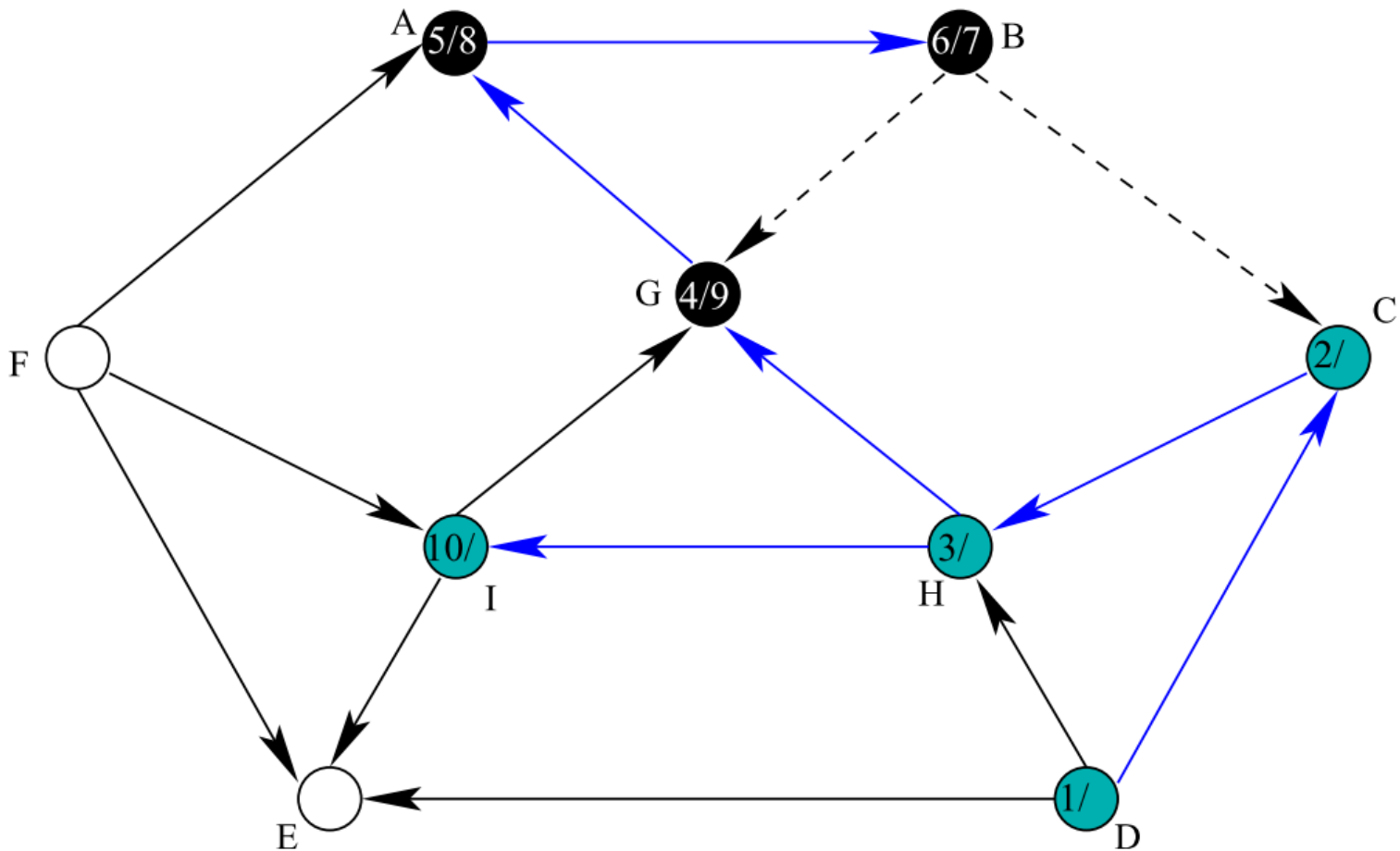


# EXEMPLO DE EXECUÇÃO



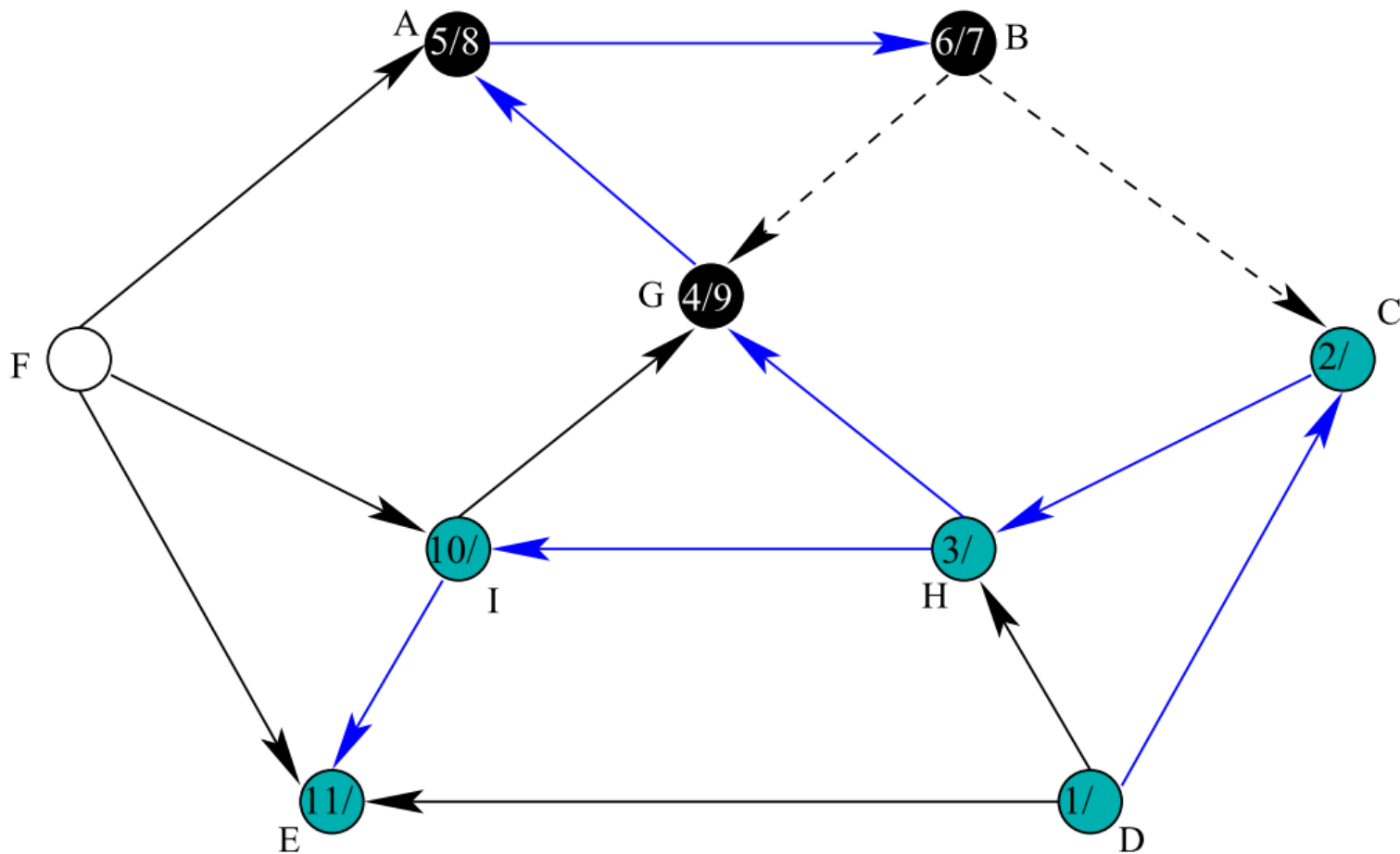


# EXEMPLO DE EXECUÇÃO





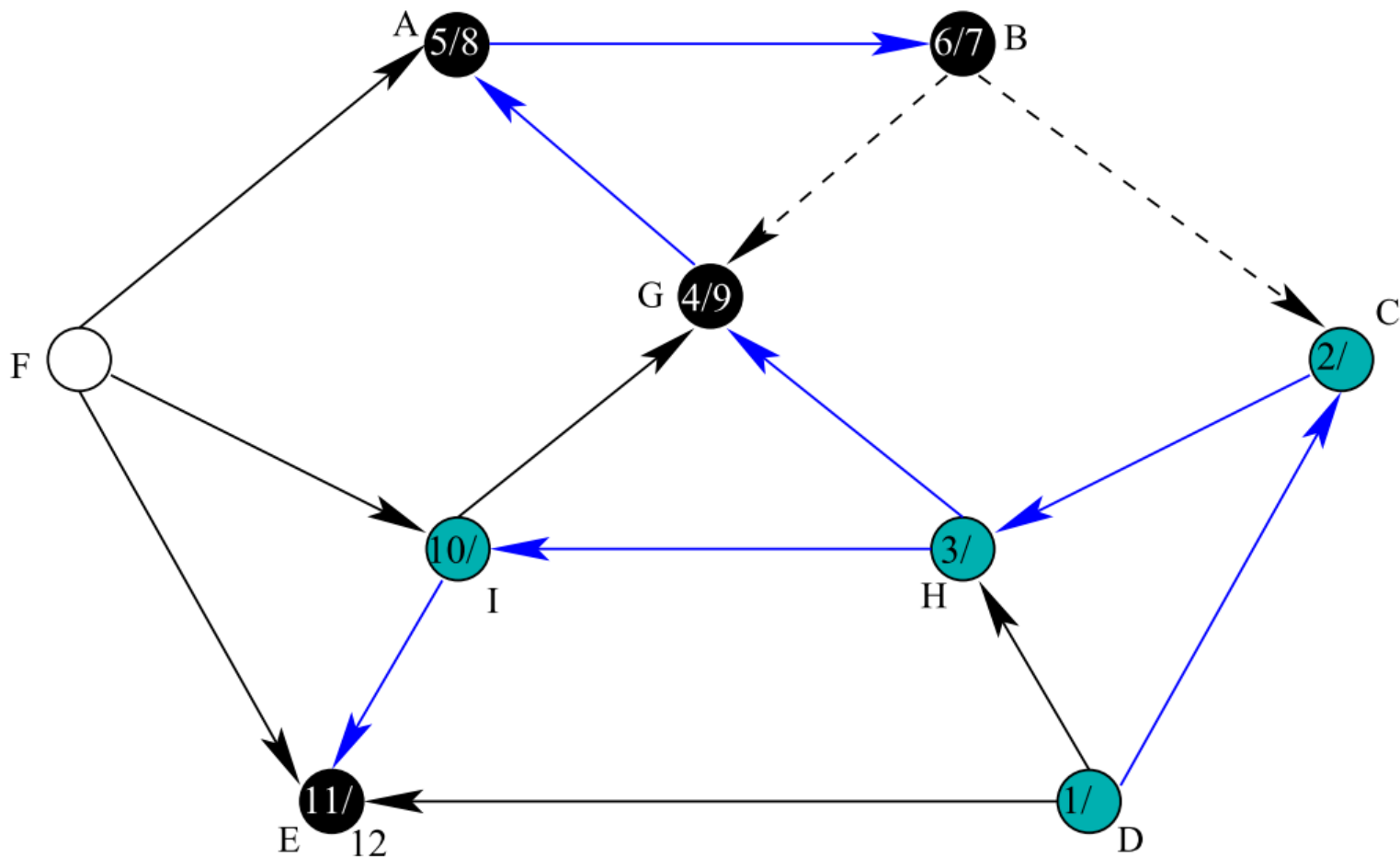
# EXEMPLO DE EXECUÇÃO





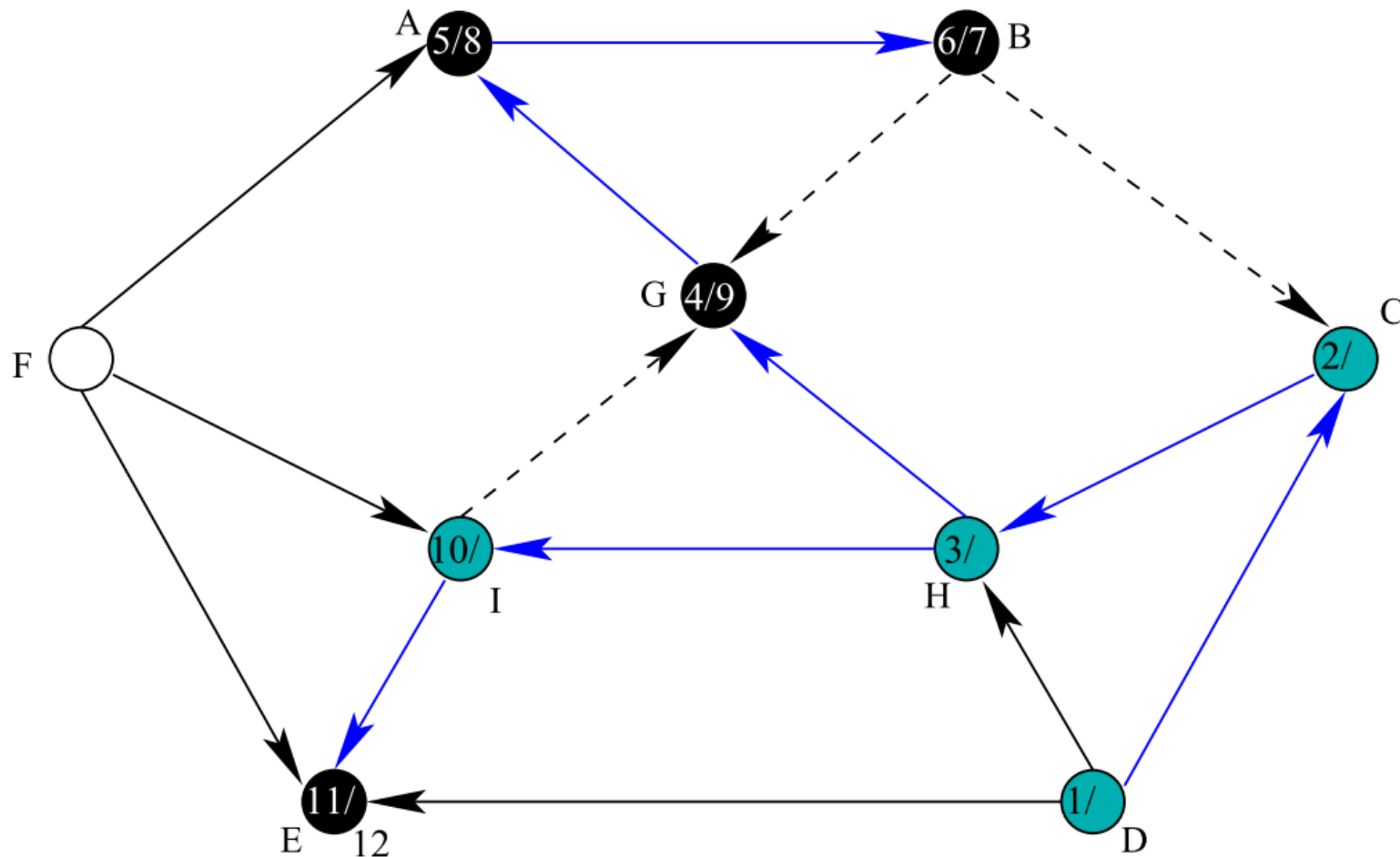


# EXEMPLO DE EXECUÇÃO



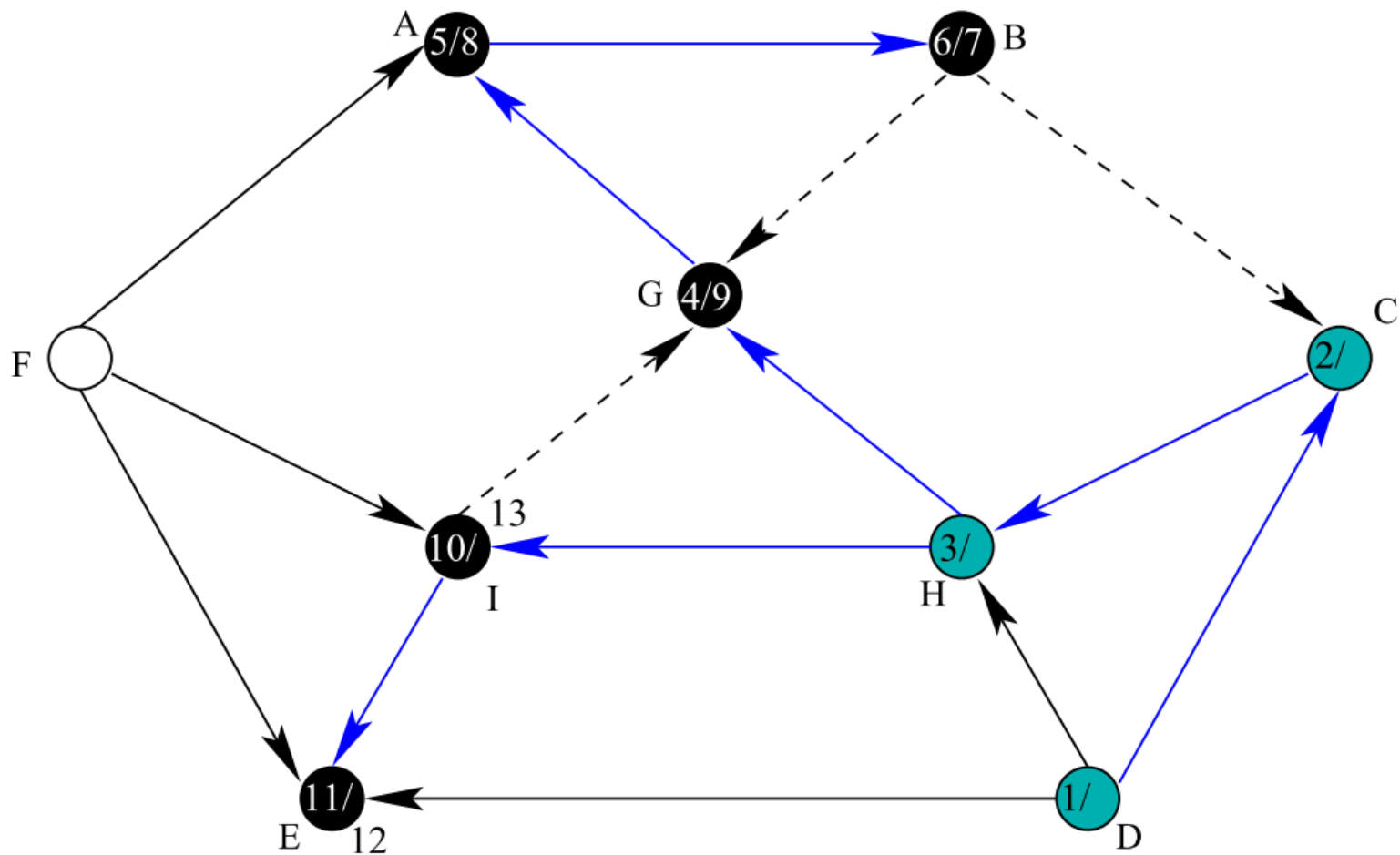


# EXEMPLO DE EXECUÇÃO



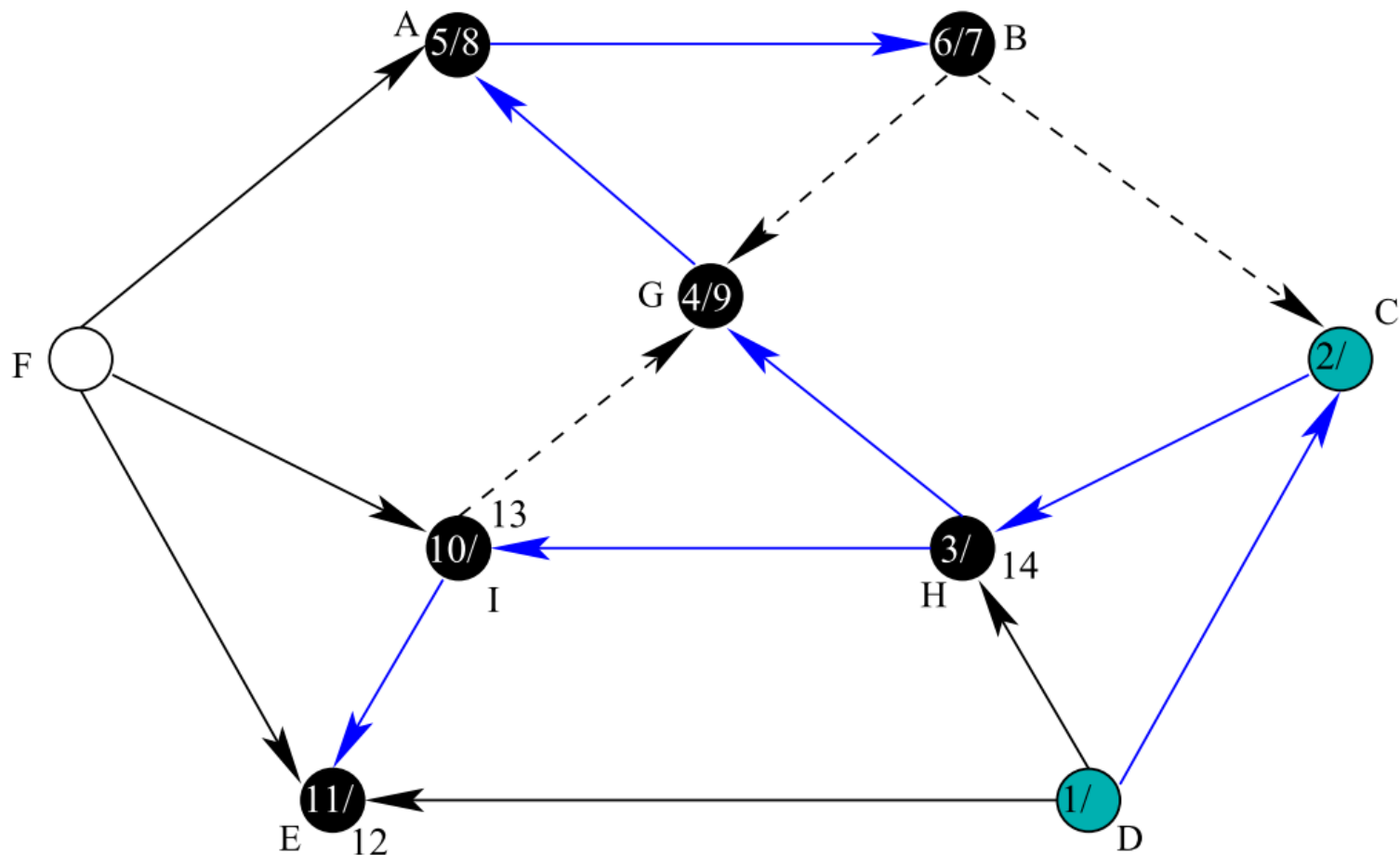


# EXEMPLO DE EXECUÇÃO



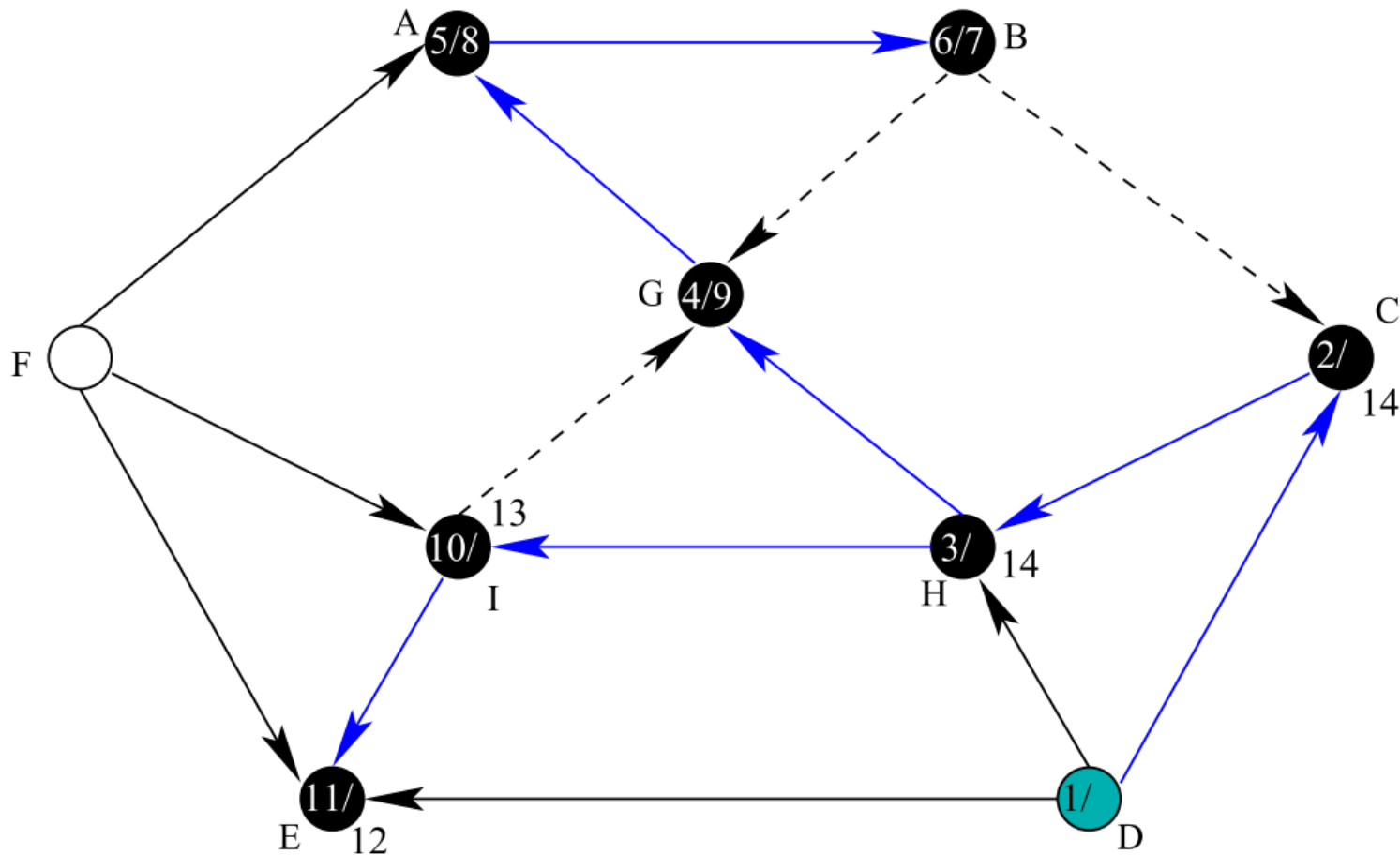


# EXEMPLO DE EXECUÇÃO

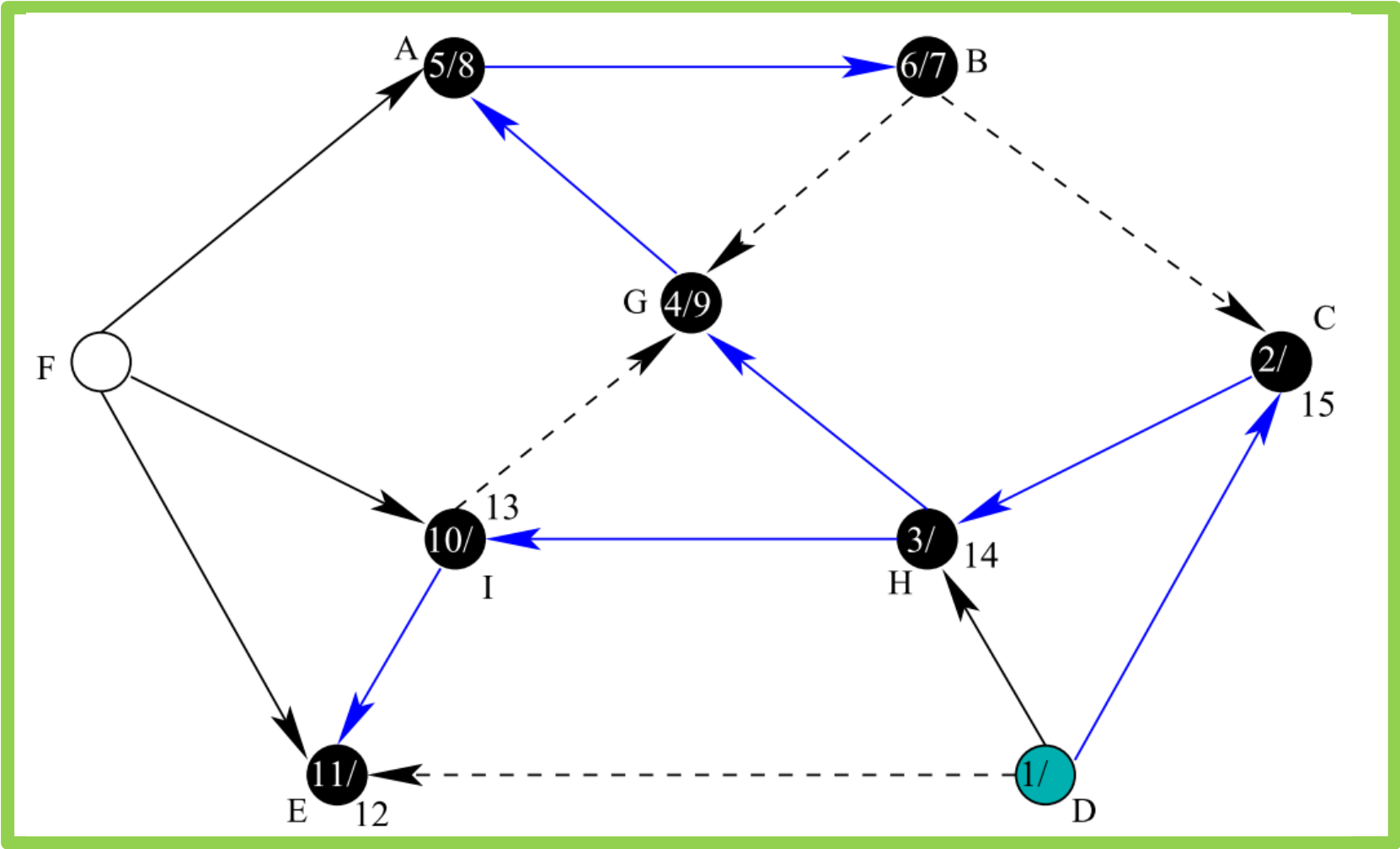




# EXEMPLO DE EXECUÇÃO

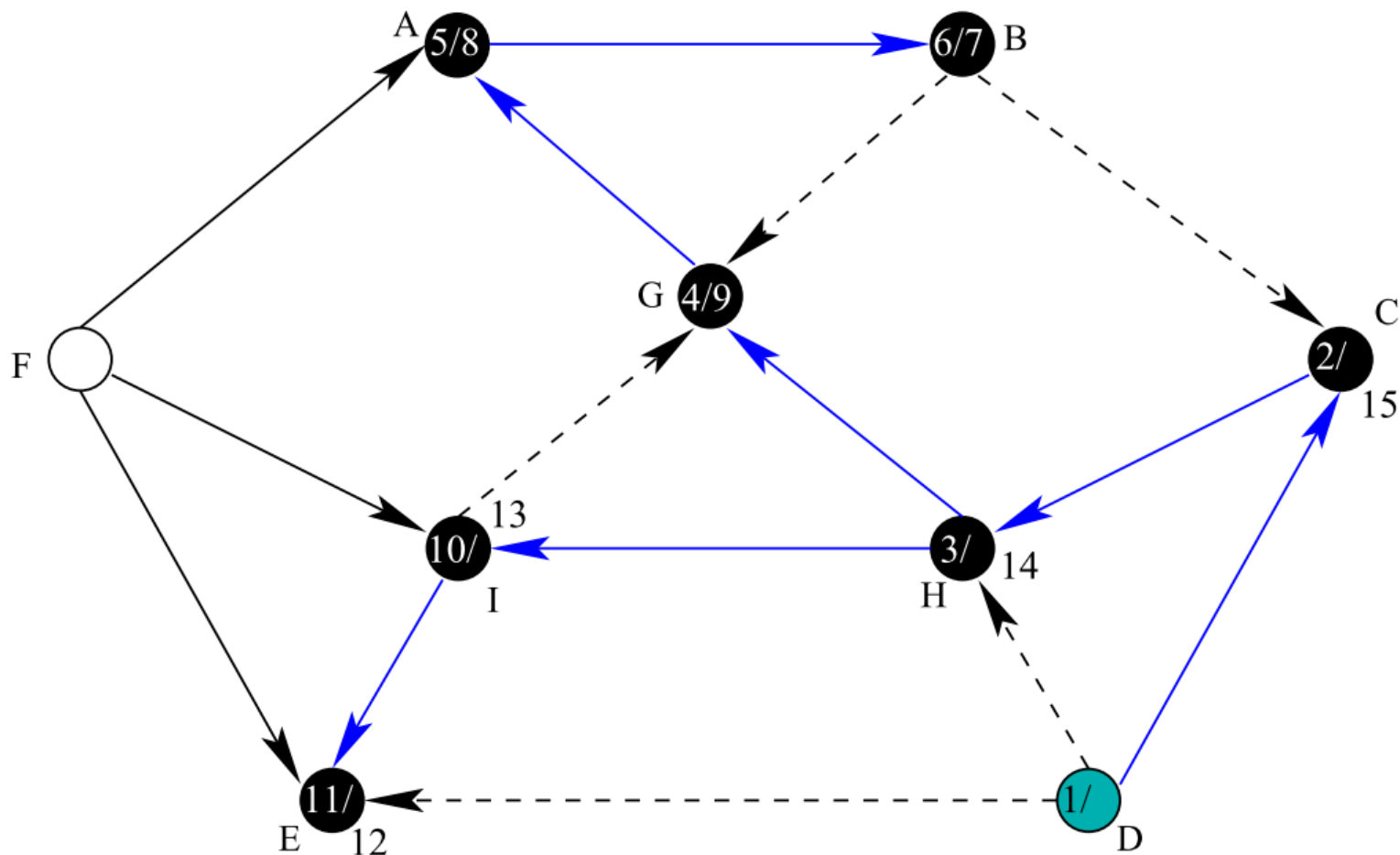


## EXEMPLO DE EXECUÇÃO



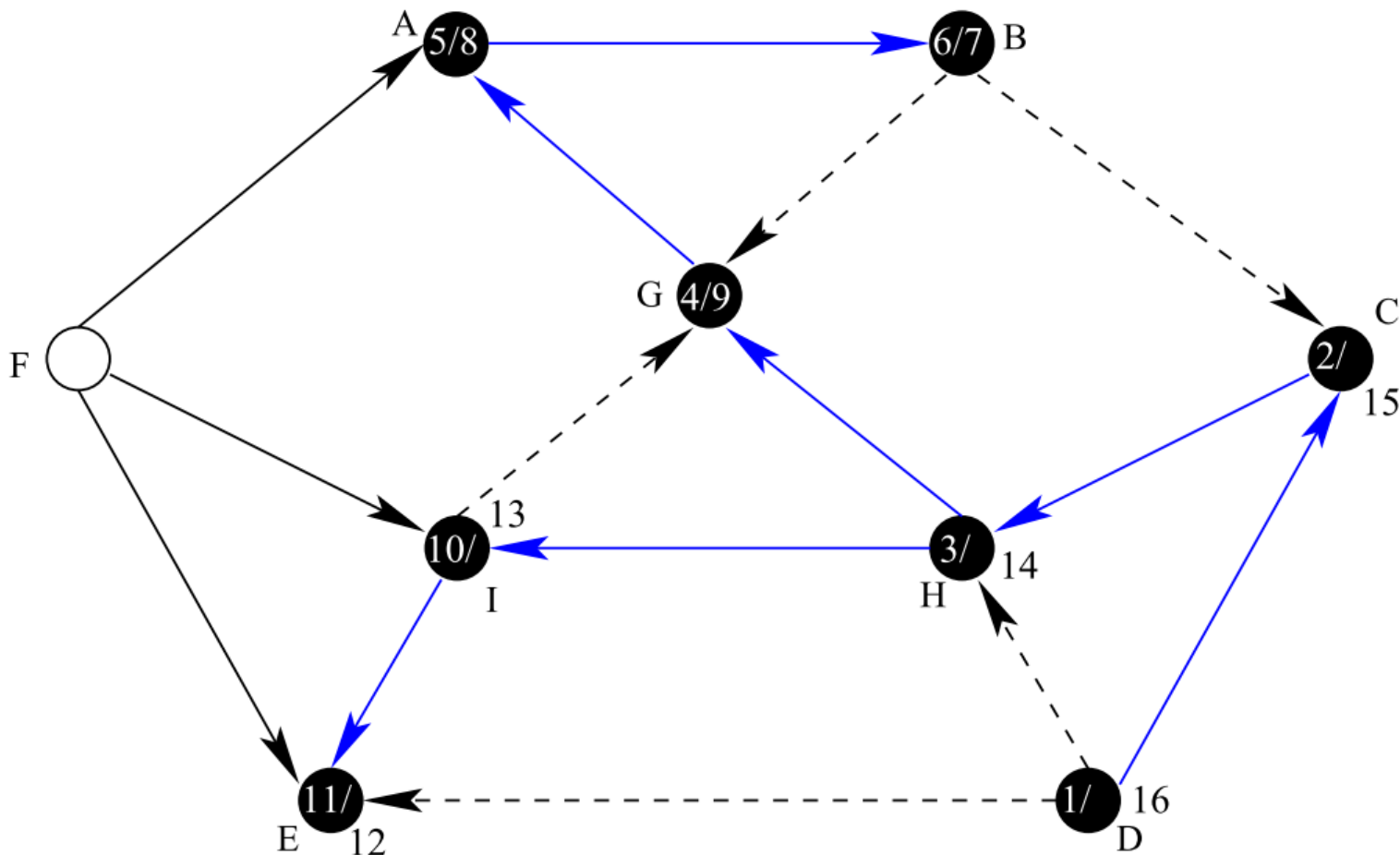


# EXEMPLO DE EXECUÇÃO





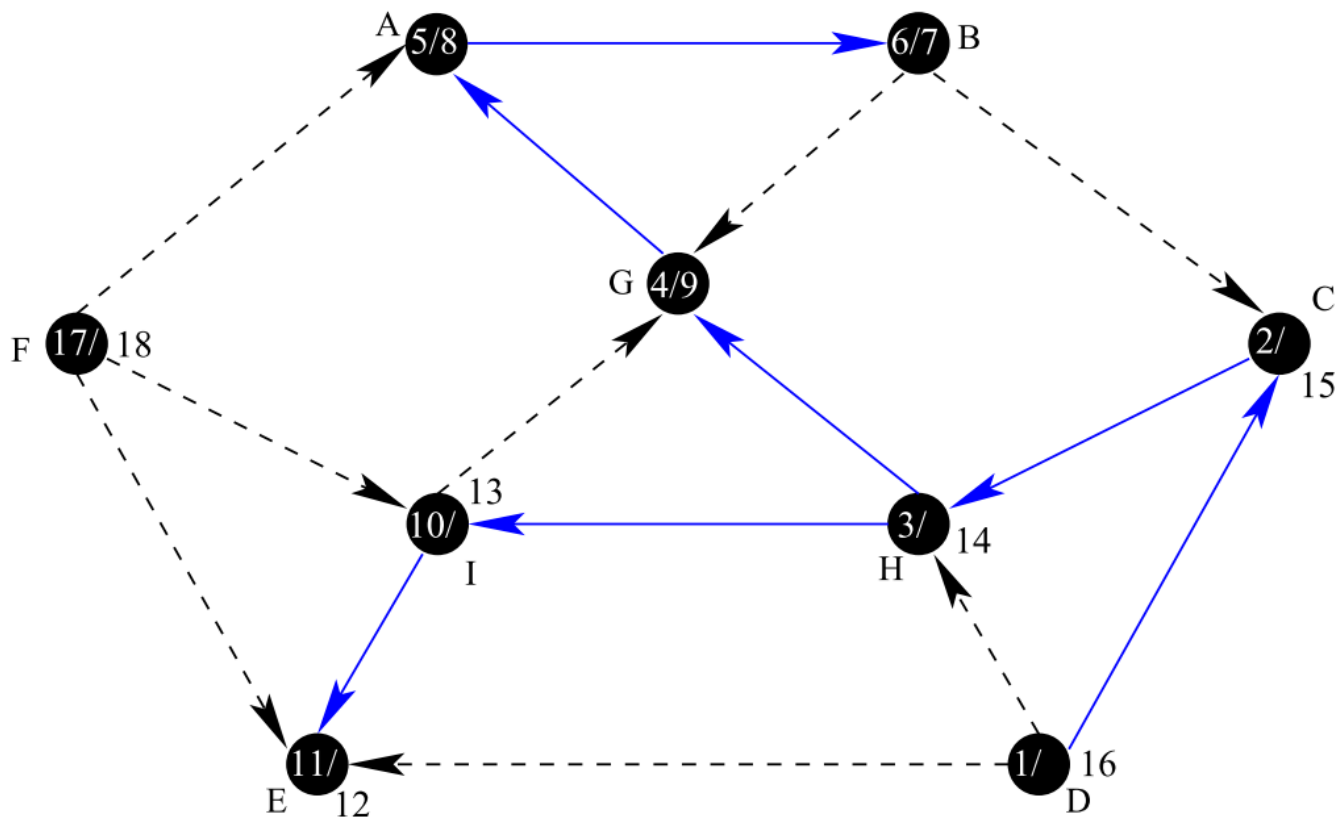
# EXEMPLO DE EXECUÇÃO







# EXEMPLO DE EXECUÇÃO



(D (C (H (G (A (B B) A) G) (I (E E) I) H) C) D) (F F)



# REFERÊNCIAS

- PEREIRA, Silvio do Lago. Estrutura de Dados Fundamentais: Conceitos e Aplicações, 12. Ed. São Paulo, Érica, 2008.
- BACKES, André Ricardo, Estrutura de dados descomplicada: em linguagem C, 1 Ed. – Rio de Janeiro: Elsevier, 2016.
- SENGER, H., Notas de Aula, Universidade de São Judas Tadeu, 1999.
- WALDEMAR Celes, Renato Cerqueira, José Lucas Rangel, Introdução a Estruturas de Dados, Editora Campus (2004).
- VELOSO, Paulo. SANTOS, Celso dos. AZEVEDO, Paulo. FURTADO, Antonio. Estrutura de dados. Rio de Janeiro: Ed. Elsevier, 1983 27ª reimpressão.
- <https://www.cs.usfca.edu/~galles/visualization/Algorithms.html>