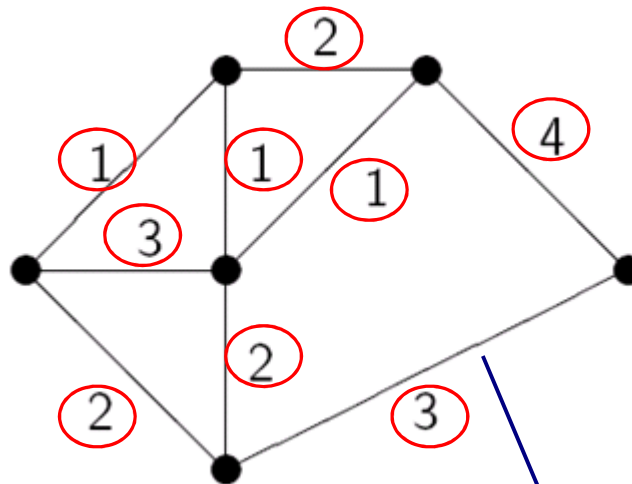


Grafos ponderados:



$$v : \mathcal{U} \rightarrow \mathbb{R}$$
$$u \mapsto v(u)$$

Função peso

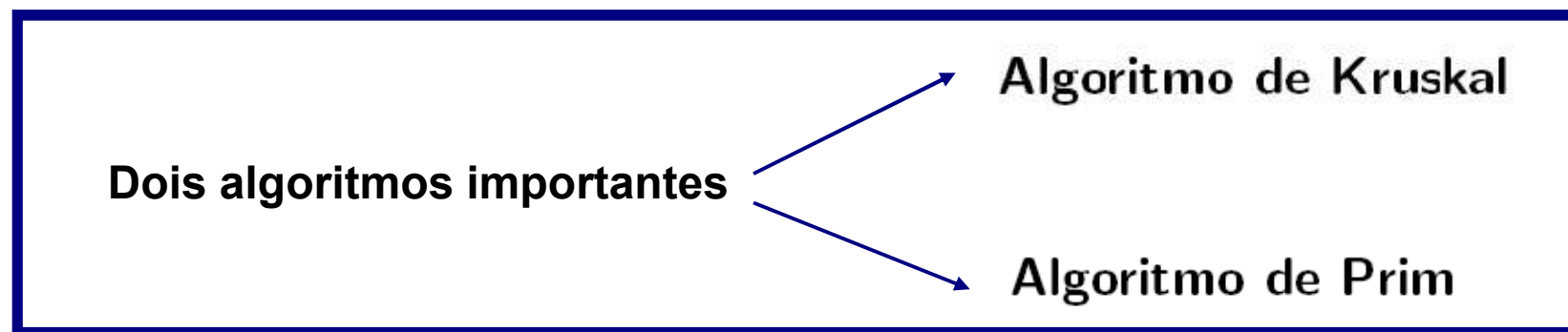
Chamamos **grafo ponderado** a um par (G, v) em que $G = (X, \mathcal{U})$ é um grafo e v é uma aplicação de $\mathcal{U}$ no conjunto dos números reais.

Se $u \in \mathcal{U}$ designa-se por **valor/peso do arco u** o número real $v(u)$ e designa-se por **valor de G** , e representa-se por $v(G)$, o número real

$$v(G) = \sum_{u \in \mathcal{U}} v(u).$$

$$V(G)=3 + 1 + 2 + = 19$$

Problema da árvore maximal de valor mínimo



Algoritmo de Kruskal

Seja (G, v) um grafo ponderado, sendo $G = (X, \mathcal{U})$ um grafo conexo com n vértices.

1º) Escolha-se um arco u_1 de G tal que

$$v(u_1) = \min_{u \in \mathcal{U}} v(u).$$

2º) Se os arcos $u_1, \dots, u_i$ já foram escolhidos, então, sendo $\mathcal{U}_i = \{u_1, \dots, u_i\}$, escolha-se um arco $u_{i+1} \in \mathcal{U}$ tal que

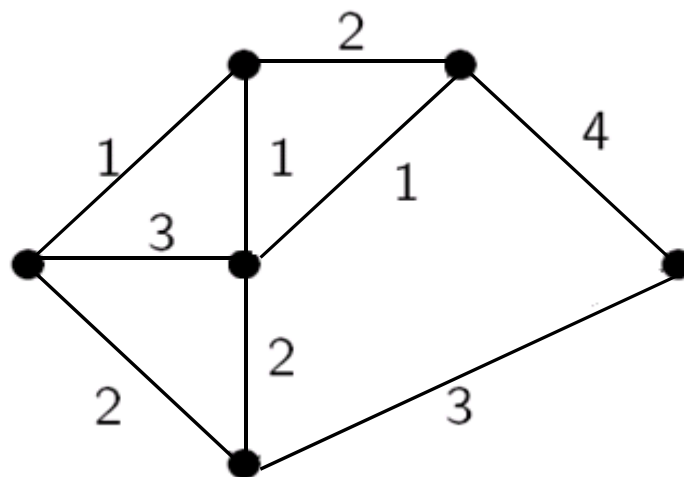
(1) $u_{i+1} \notin \mathcal{U}_i$

(2) $G' = (X, \mathcal{U}_i \cup \{u_{i+1}\})$ não tem ciclos

(3) u_{i+1} é de entre os arcos que verificam as condições (1) e (2), um com valor mínimo.

3º) Se já foram escolhidos $n - 1$ arcos, então o algoritmo termina. Caso contrário, repita-se 2º).

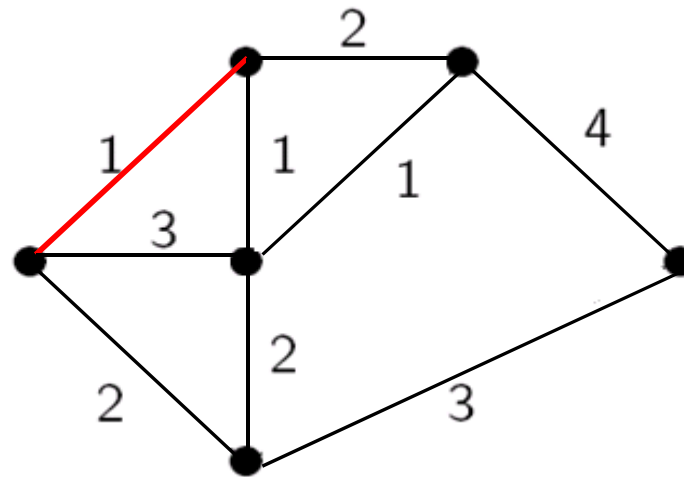
Exemplo 1: Consideremos o seguinte grafo ponderado **Algoritmo de Kruskal**



Escolha do melhor
arco de entre os que
faltam escolher

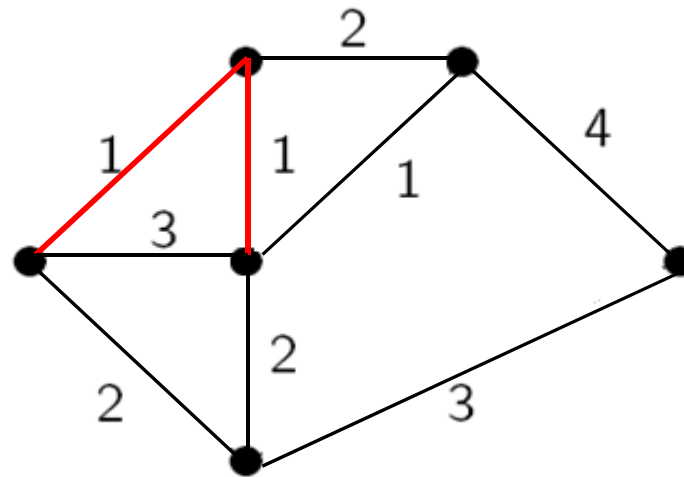
Calculemos, usando o Algoritmo de Kruskal, uma árvore maximal de valor mínimo.

Exemplo 1: Consideremos o seguinte grafo ponderado



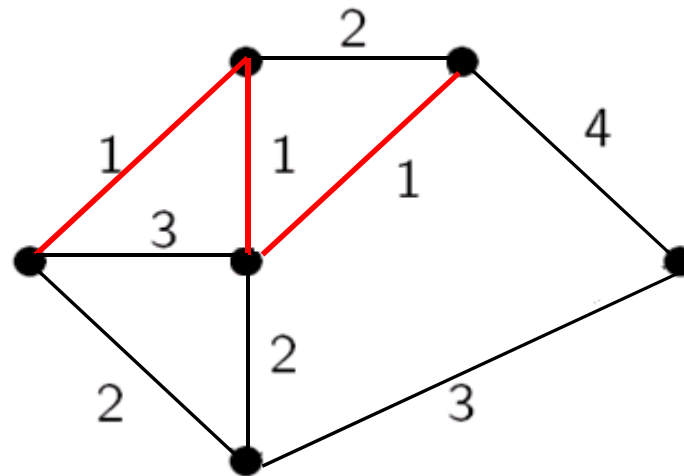
Calculemos, usando o Algoritmo de Kruskal, uma árvore maximal de valor mínimo.

Exemplo 1: Consideremos o seguinte grafo ponderado



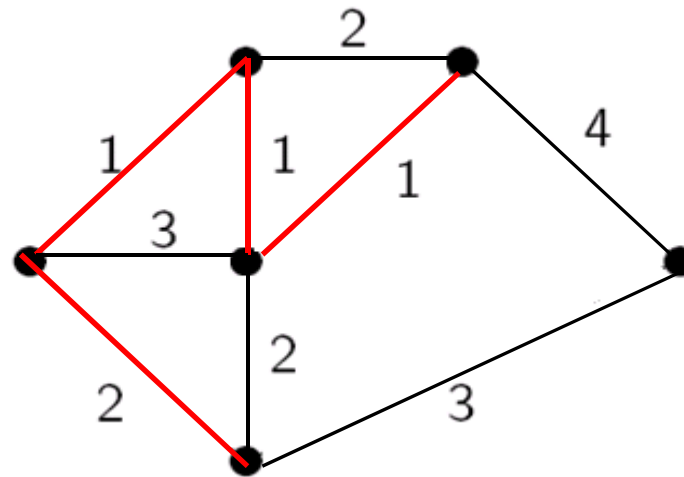
Calculemos, usando o Algoritmo de Kruskal, uma árvore maximal de valor mínimo.

Exemplo 1: Consideremos o seguinte grafo ponderado



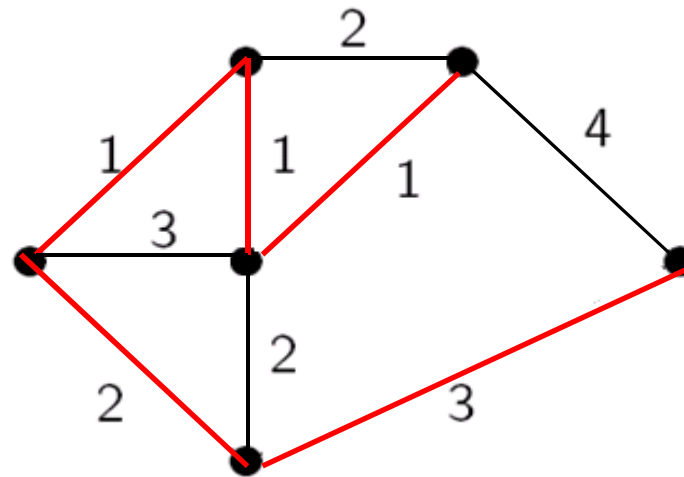
Calculemos, usando o Algoritmo de Kruskal, uma árvore maximal de valor mínimo.

Exemplo 1: Consideremos o seguinte grafo ponderado



Calculemos, usando o Algoritmo de Kruskal, uma árvore maximal de valor mínimo.

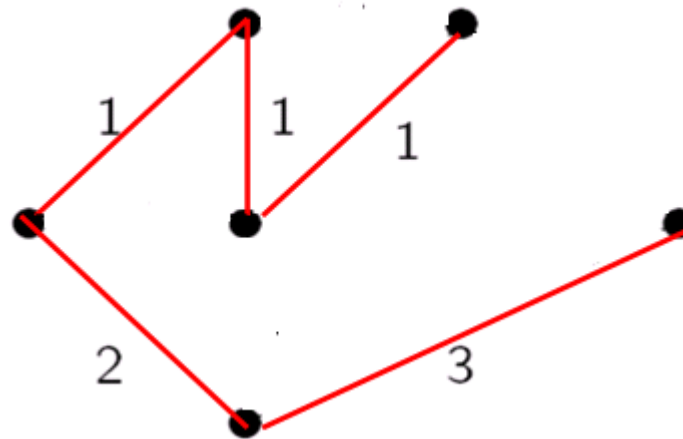
Exemplo 1: Consideremos o seguinte grafo ponderado



5 arcos escolhidos

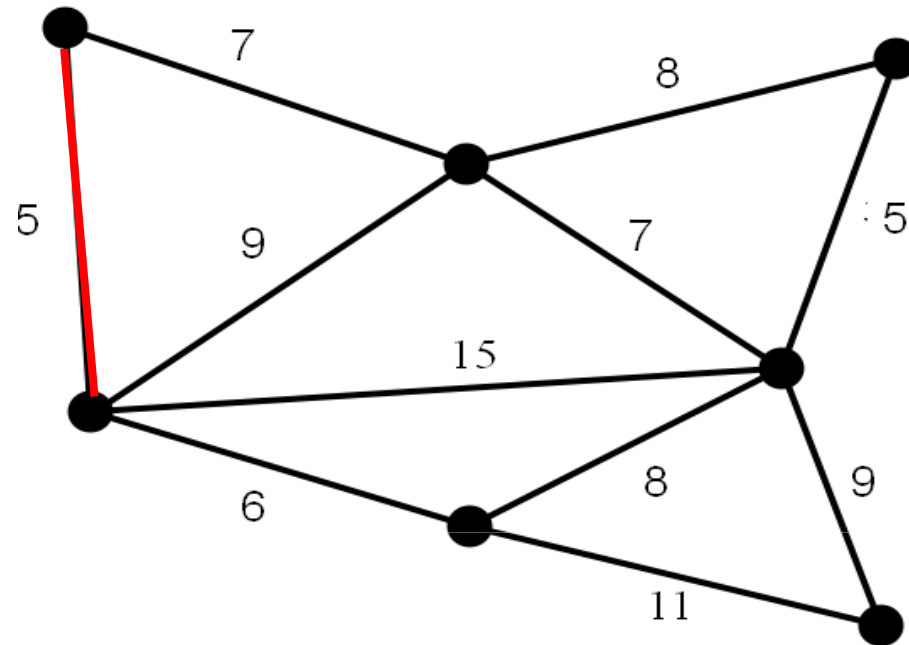


Calculemos, usando o Algoritmo de Kruskal, uma árvore maximal de valor mínimo.



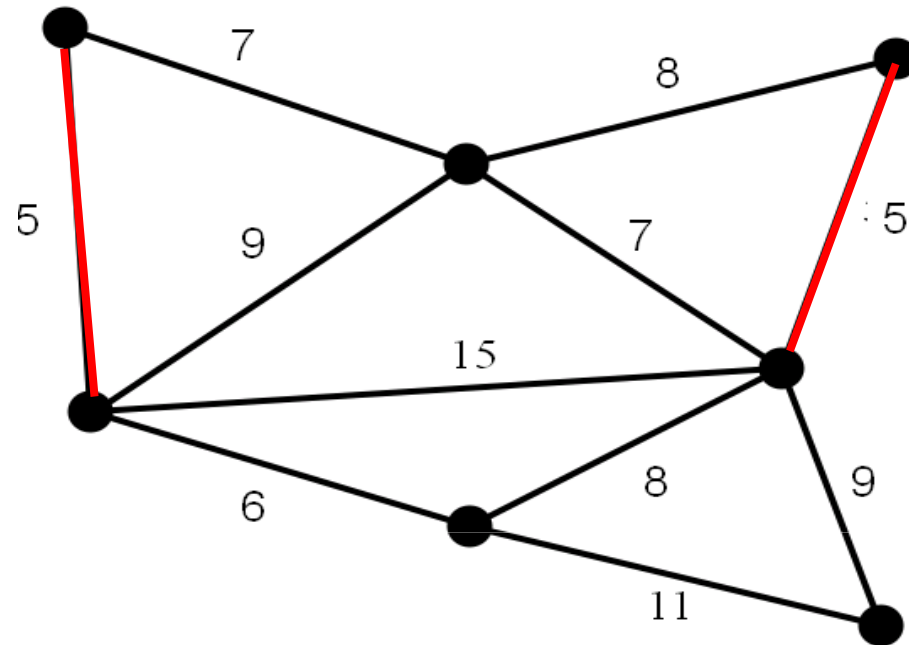
Valor = 8

Exemplo 2: Consideremos o seguinte grafo ponderado



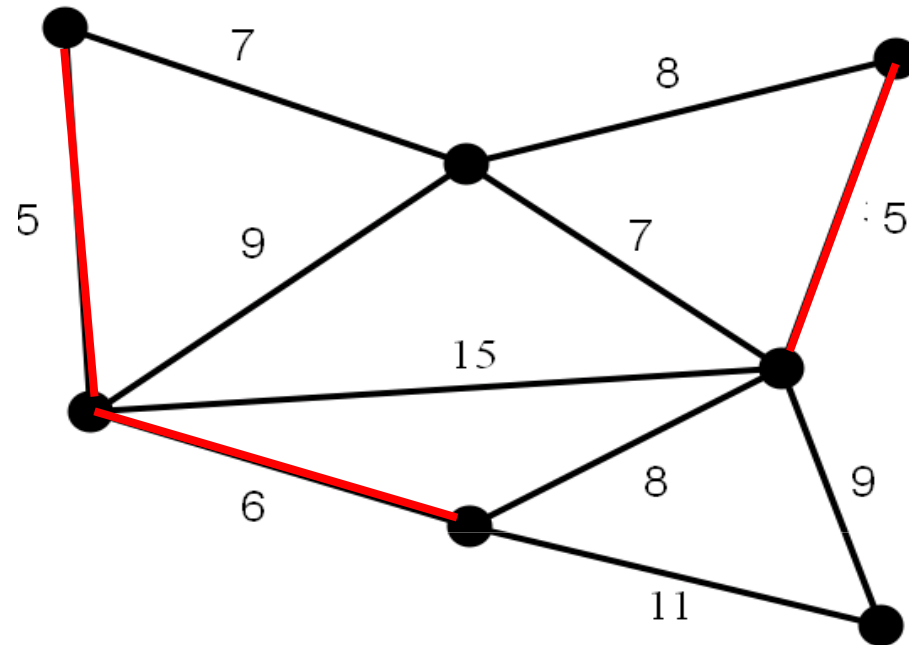
Calculemos, usando o Algoritmo de Kruskal, uma árvore maximal de valor mínimo.

Exemplo 2: Consideremos o seguinte grafo ponderado



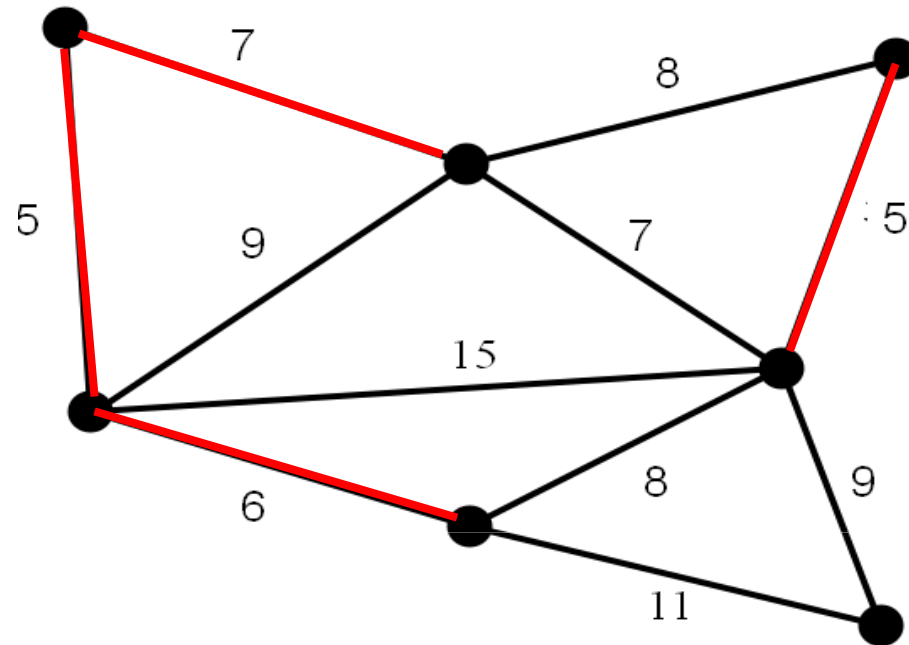
Calculemos, usando o Algoritmo de Kruskal, uma árvore maximal de valor mínimo.

Exemplo 2: Consideremos o seguinte grafo ponderado



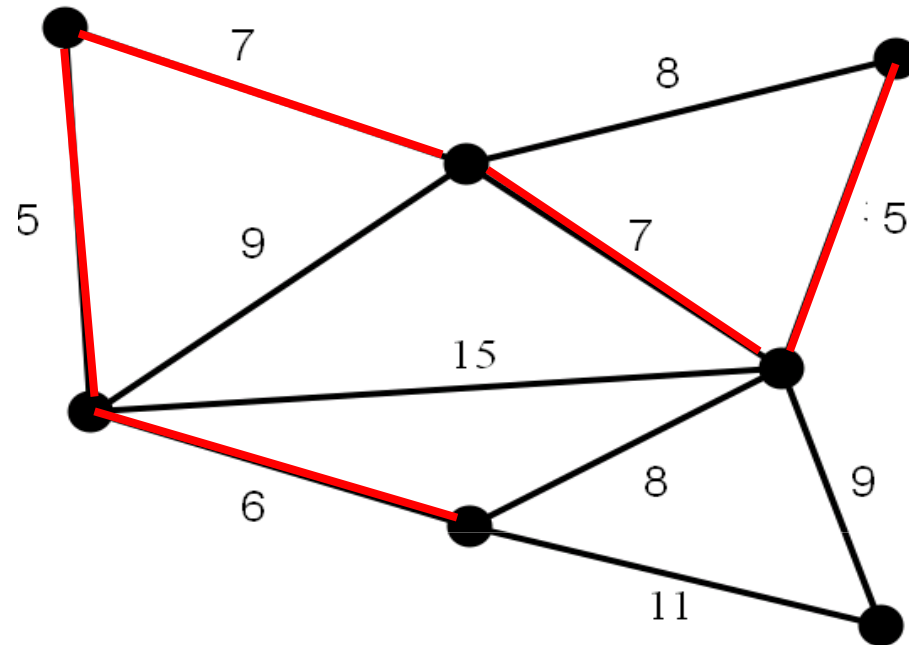
Calculemos, usando o Algoritmo de Kruskal, uma árvore maximal de valor mínimo.

Exemplo 2: Consideremos o seguinte grafo ponderado



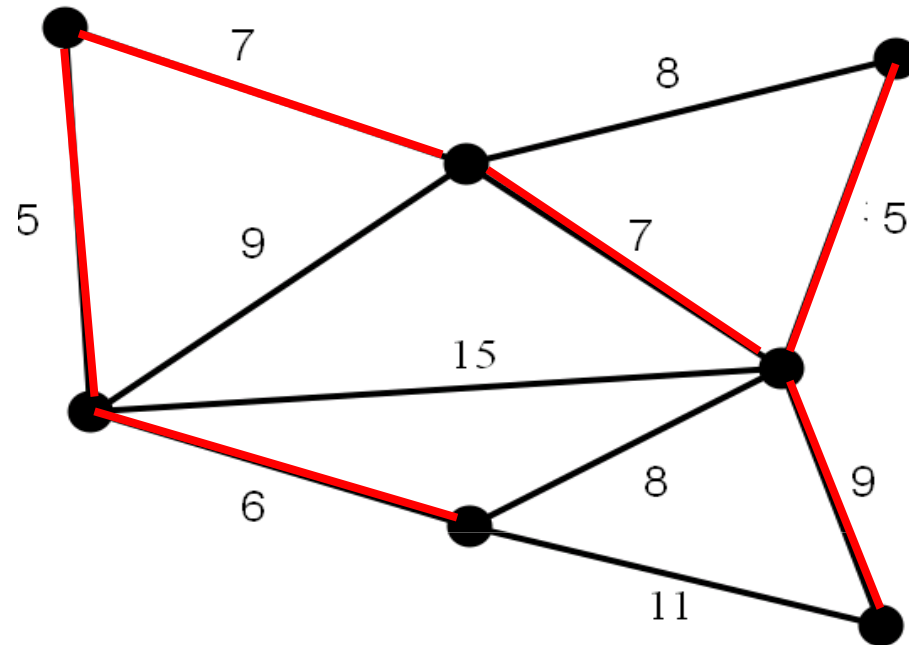
Calculemos, usando o Algoritmo de Kruskal, uma árvore maximal de valor mínimo.

Exemplo 2: Consideremos o seguinte grafo ponderado



Calculemos, usando o Algoritmo de Kruskal, uma árvore maximal de valor mínimo.

Exemplo 2: Consideremos o seguinte grafo ponderado



6 arcos escolhidos



Valor = 39

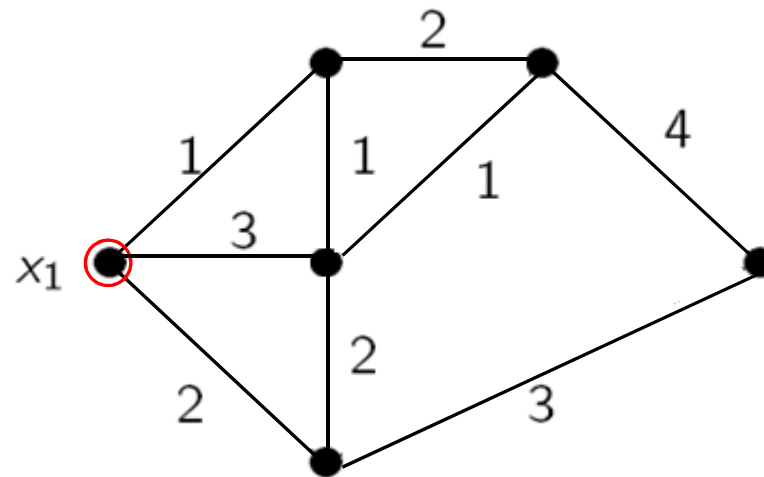
Calculemos, usando o Algoritmo de Kruskal, uma árvore maximal de valor mínimo.

Algoritmo de Prim

Seja (G, v) um grafo ponderado, sendo $G = (X, \mathcal{U})$ um grafo conexo com n vértices.

- 1º) Considere-se um vértice arbitrário de G , que designamos por x_1 .
- 2º) Escolha-se um arco u_1 de G , incidente em x_1 , que tenha valor mínimo.
- 3º) Se os arcos $u_1, \dots, u_i$ já foram escolhidos e sendo $X_i = \{x_1, \dots, x_{i+1}\}$ o conjunto de vértices formado pela suas extremidades, escolha-se um arco u_{i+1} de valor mínimo entre os arcos $\{x_j, x_{i+2}\}$ de G tal que $x_j \in X_i$ e $x_{i+2} \notin X_i$ (i.e. u_{i+1} é um arco de valor mínimo entre os arcos ainda não escolhidos com precisamente uma extremidade em X_i).
- 4º) Se já foram escolhidos $n - 1$ arcos, então o algoritmo termina. Caso contrário, repita-se 3º).

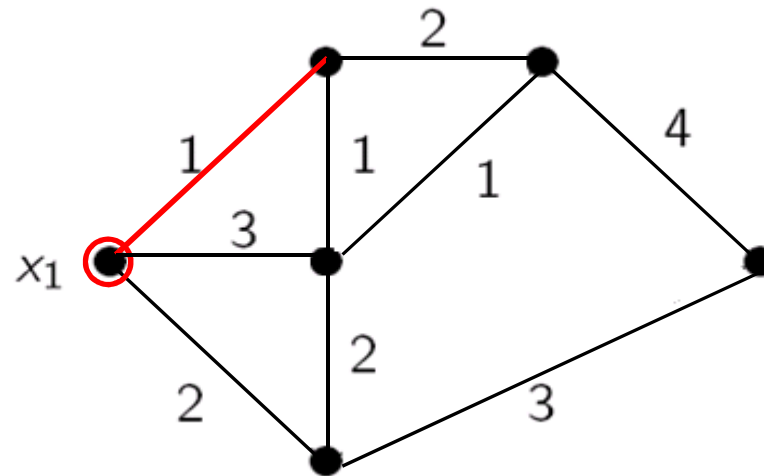
Exemplo 1: Com o grafo ponderado do exemplo anterior, **Algoritmo de Prim**



↑
Evolui pelo princípio do
vizinho mais próximo

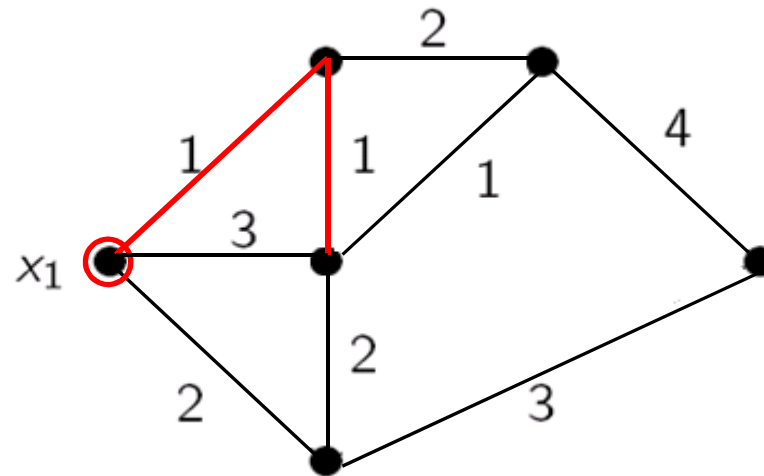
calculemos uma árvore maximal de valor mínimo, mas utilizando o Algoritmo de Prim e partindo do vértice x_1 .

Exemplo 1: Com o grafo ponderado do exemplo anterior,



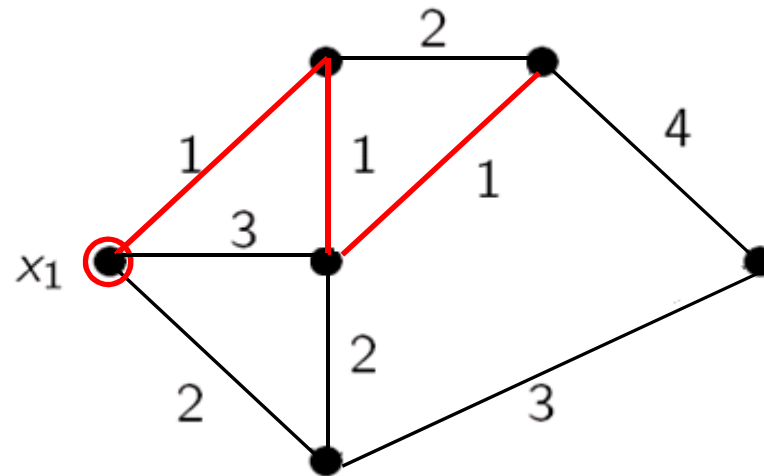
calculemos uma árvore maximal de valor mínimo, mas utilizando o Algoritmo de Prim e partindo do vértice x_1 .

Exemplo 1: Com o grafo ponderado do exemplo anterior,



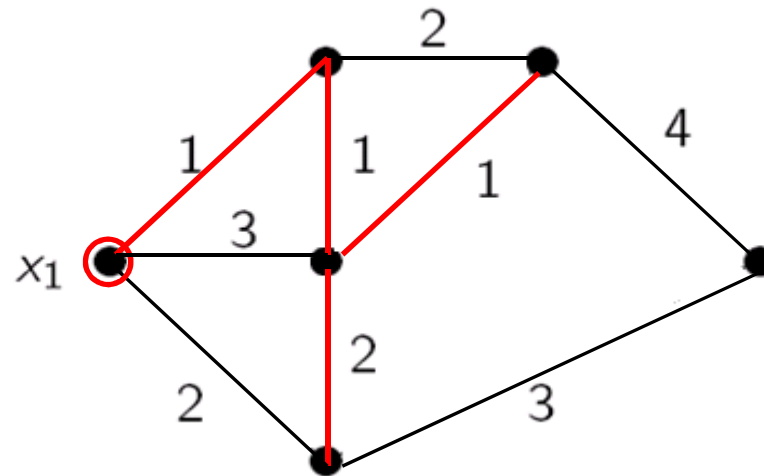
calculemos uma árvore maximal de valor mínimo, mas utilizando o Algoritmo de Prim e partindo do vértice x_1 .

Exemplo 1: Com o grafo ponderado do exemplo anterior,



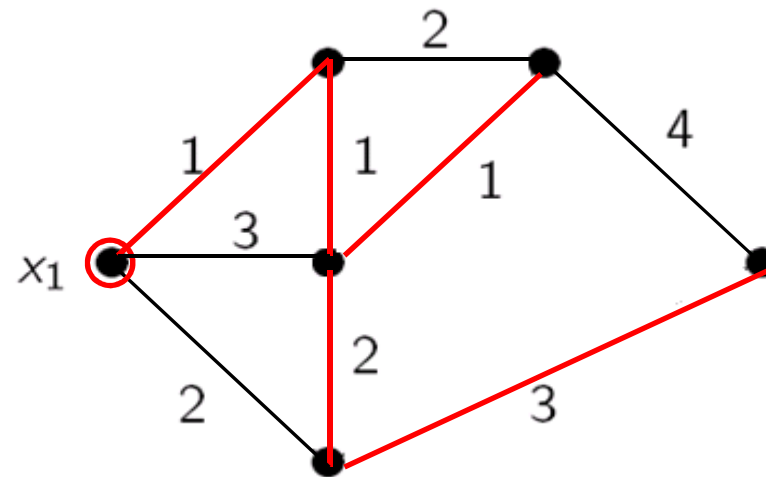
calculemos uma árvore maximal de valor mínimo, mas utilizando o Algoritmo de Prim e partindo do vértice x_1 .

Exemplo 1: Com o grafo ponderado do exemplo anterior,



calculemos uma árvore maximal de valor mínimo, mas utilizando o Algoritmo de Prim e partindo do vértice x_1 .

Exemplo 1: Com o grafo ponderado do exemplo anterior,



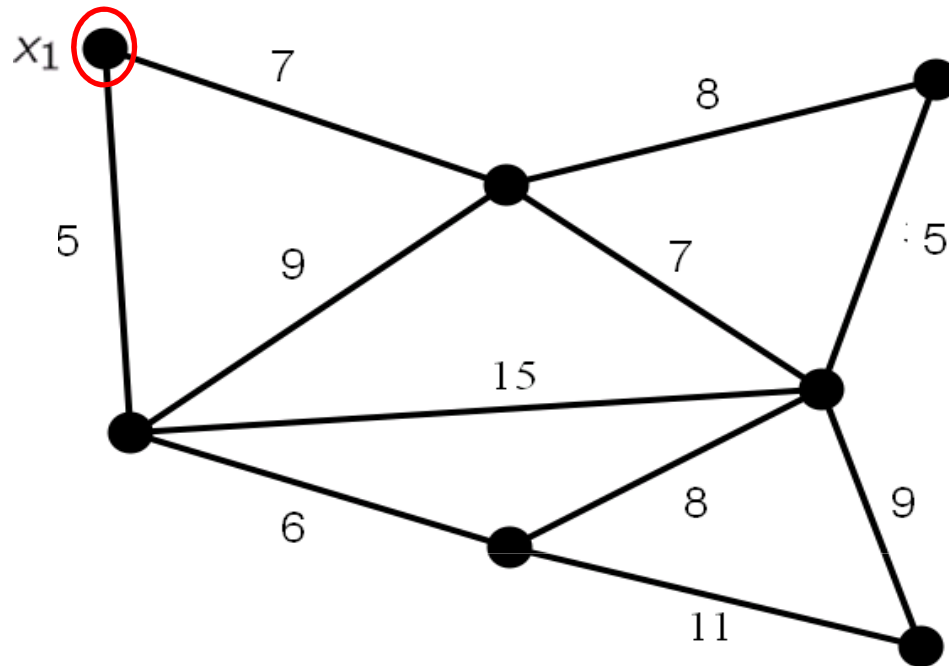
5 arcos escolhidos



Valor = 8

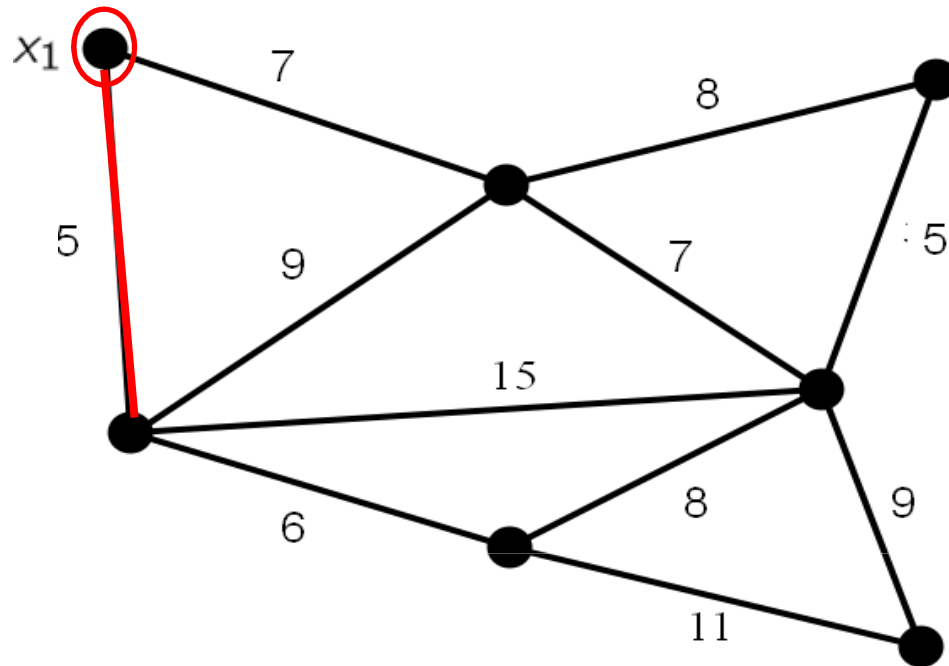
calculemos uma árvore maximal de valor mínimo, mas utilizando o Algoritmo de Prim e partindo do vértice x_1 .

Exemplo 2: Consideremos o seguinte grafo ponderado



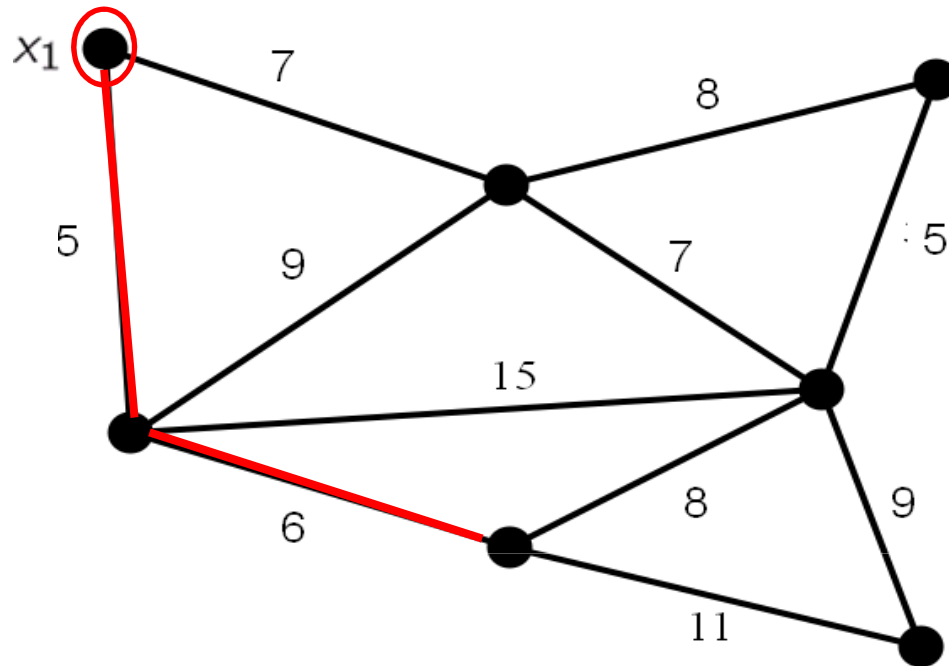
calculemos uma árvore maximal de valor mínimo, mas utilizando o Algoritmo de Prim e partindo do vértice x_1 .

Exemplo 2: Consideremos o seguinte grafo ponderado



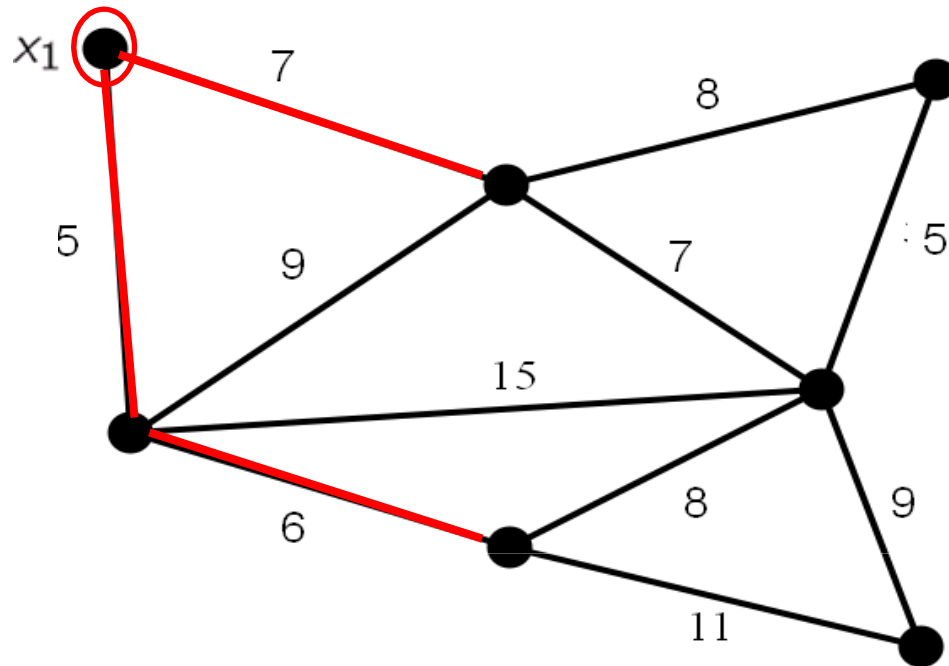
calculemos uma árvore maximal de valor mínimo, mas utilizando o Algoritmo de Prim e partindo do vértice x_1 .

Exemplo 2: Consideremos o seguinte grafo ponderado



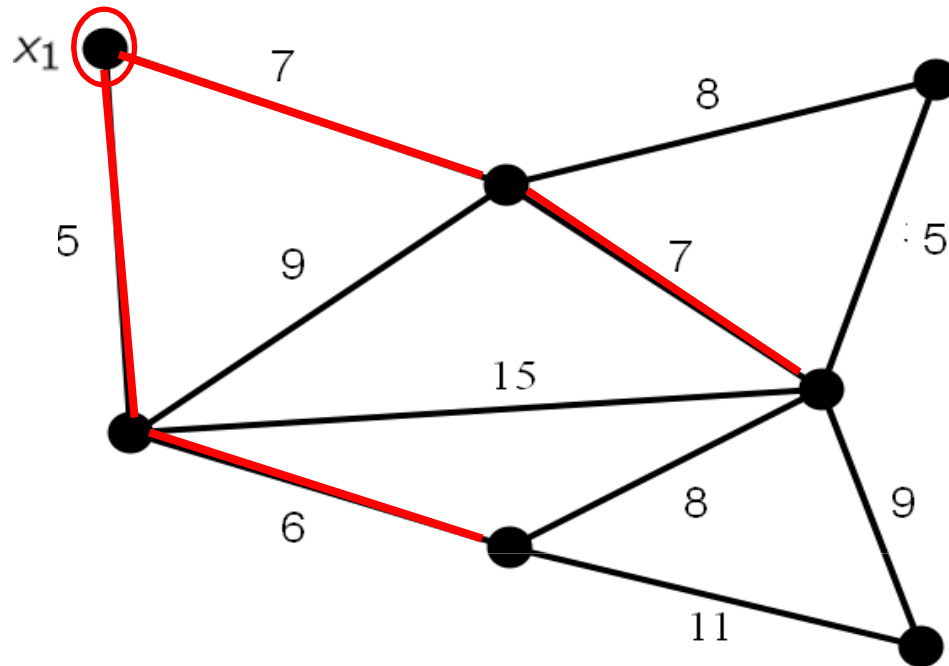
calculemos uma árvore maximal de valor mínimo, mas utilizando o Algoritmo de Prim e partindo do vértice x_1 .

Exemplo 2: Consideremos o seguinte grafo ponderado



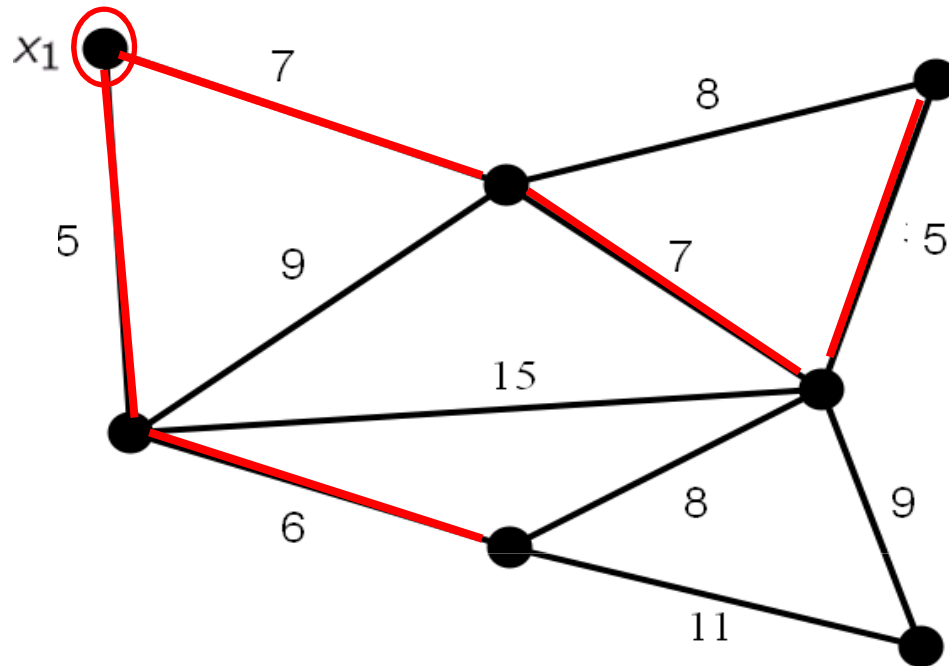
calculemos uma árvore maximal de valor mínimo, mas utilizando o Algoritmo de Prim e partindo do vértice x_1 .

Exemplo 2: Consideremos o seguinte grafo ponderado



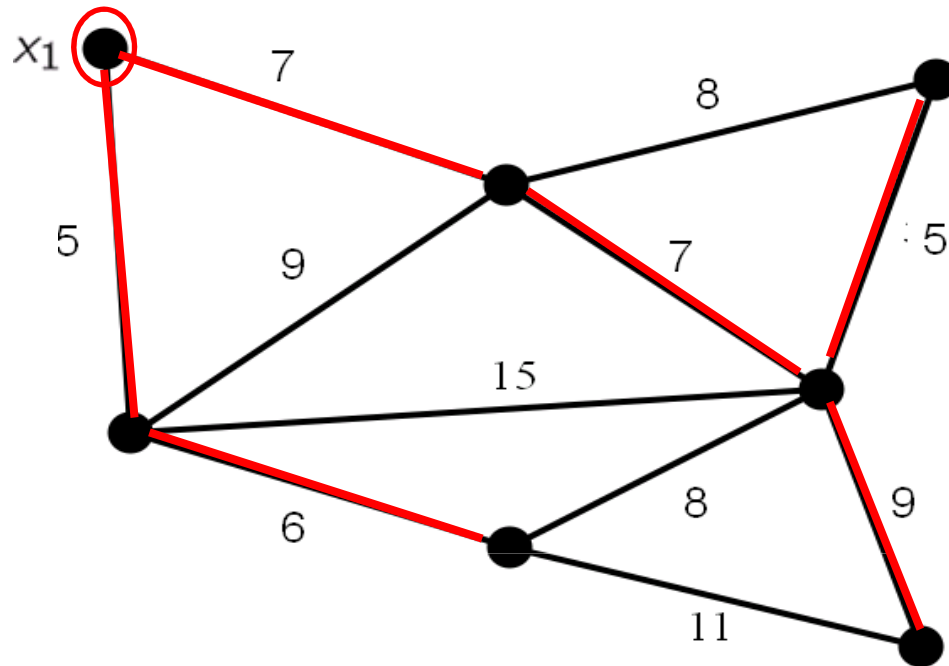
calculemos uma árvore maximal de valor mínimo, mas utilizando o Algoritmo de Prim e partindo do vértice x_1 .

Exemplo 2: Consideremos o seguinte grafo ponderado



calculemos uma árvore maximal de valor mínimo, mas utilizando o Algoritmo de Prim e partindo do vértice x_1 .

Exemplo 2: Consideremos o seguinte grafo ponderado



6 arcos escolhidos

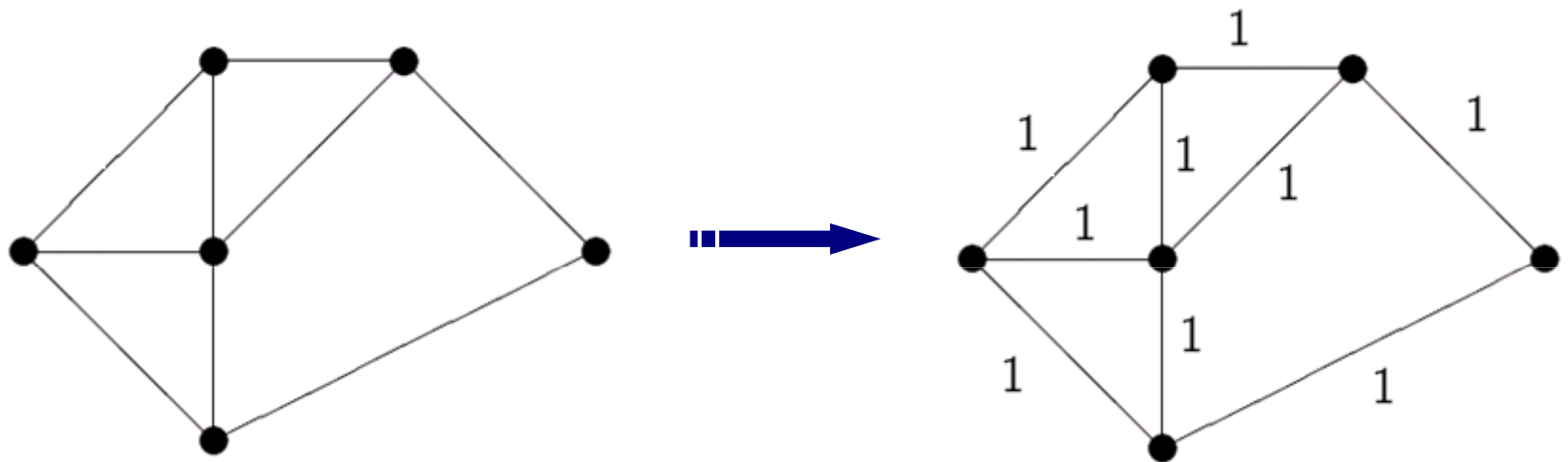


Valor = 39

calculemos uma árvore maximal de valor mínimo, mas utilizando o Algoritmo de Prim e partindo do vértice x_1 .

Observação:

- Se temos um grafo simples conexo e queremos uma sua árvore maximal, podemos usar qualquer um dos algoritmos anteriores bastando para isso, atribuir o mesmo valor a todos os arcos do grafo.



- Em geral, em termos de implementação em computador, o Algoritmo de Prim é mais rápido do que o de Kruskal.

Problema da cadeia mais curta

Sejam (G, v) um grafo conexo ponderado e x e y dois vértices de G . O problema da cadeia mais curta consiste em determinar uma cadeia $x - y$

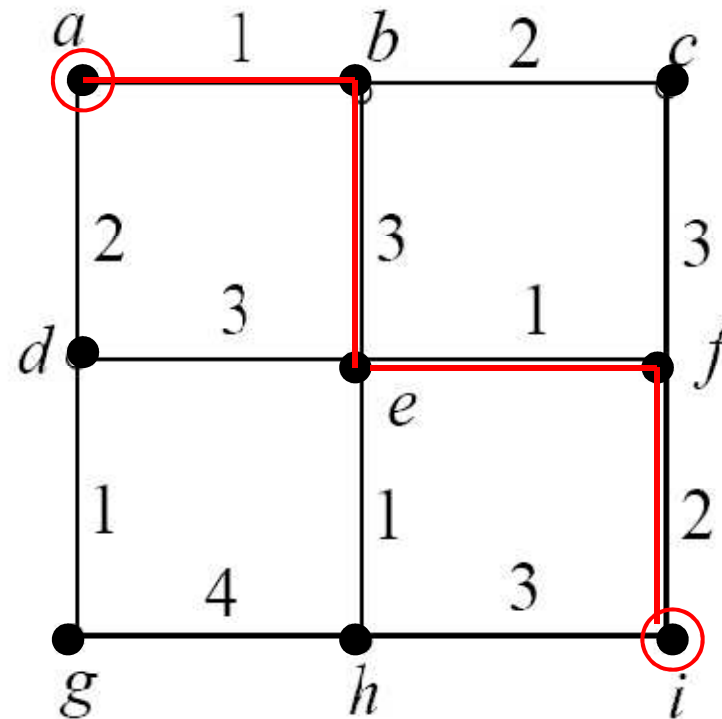
$$L : x = x_0, u_1, x_1, u_2, \dots, u_k, x_k = y$$

por forma que o valor de L , i.e.

$$v(L) = \sum_{i=1}^k v(u_i)$$

(soma dos valores dos arcos de L), seja o mínimo possível. Neste caso, dizemos que L é uma **cadeia $x - y$** (de valor) **mínima**.

Exemplo: Consideremos o seguinte grafo ponderado:



Uma breve análise permite-nos determinar facilmente que

$$L: a \rightarrow b \rightarrow e \rightarrow f \rightarrow i$$

é uma cadeia mínima do vértice a para o vértice i (cujo valor é igual a 7).

No entanto, para grafos ponderados de ordens e tamanhos maiores, torna-se necessário aplicar um método sistemático para encontrar uma tal cadeia. Enunciamos a seguir um algoritmo para resolver este problema.

Algoritmo da Cadeia mais Curta (Dijkstra)

Seja (G, v) um grafo conexo ponderado, em que v toma valores não negativos, e sejam x e y dois vértices distintos de G . Designemos x por vértice inicial e y por vértice final.

PASSO 1.

- ① Atribuir a x uma *etiqueta definitiva* igual a 0;
- ② Atribuir a cada vértice x' adjacente a x uma *etiqueta temporária* igual ao valor do arco correspondente, i.e. igual a $v(\{x, x'\})$;
- ③ Sendo ε a menor das etiquetas temporárias acabadas de atribuir, para cada vértice z com etiqueta temporária igual a ε , atribuir a z uma etiqueta definitiva igual a ε .

PASSO 2.

Se y tem atribuída uma etiqueta definitiva, **TERMINAR**, caso contrário ir para o PASSO 3.

PASSO 3.

- 1 Para cada vértice z ao qual se acabou de atribuir uma etiqueta definitiva, $\bar{\varepsilon}$, e para cada vértice z' adjacente a z , atribuir a z' uma etiqueta temporária igual a

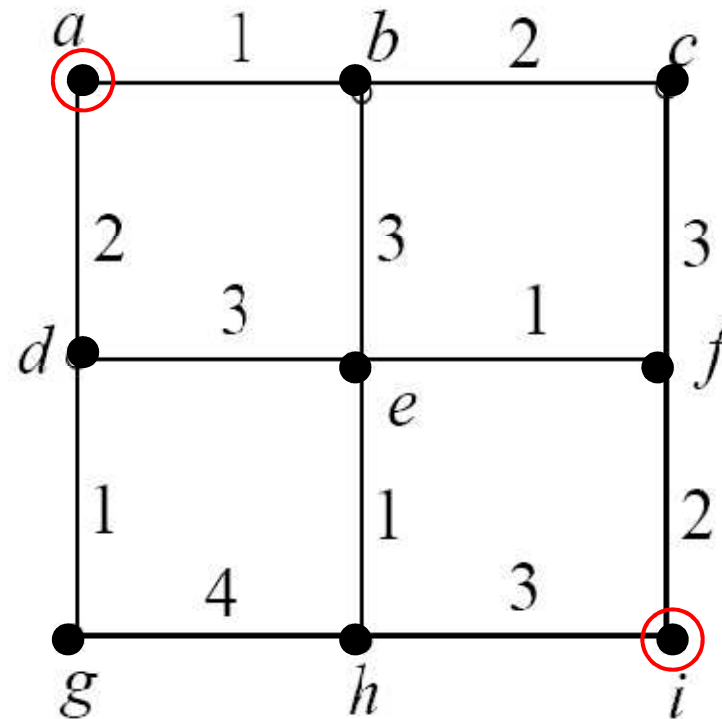
$$\bar{\varepsilon} + v(\{z, z'\}),$$

excepto se z' já possui uma etiqueta de valor inferior;

- 2 Sendo ε a menor das etiquetas temporárias de **todos** os vértices no grafo com etiquetas temporárias atribuídas (não só aos que acabámos de atribuir), para cada vértice z (no grafo) com etiqueta temporária igual a ε , atribuir a z uma etiqueta definitiva igual a ε ;
- 3 Ir para o PASSO 2.

Algoritmo da Cadeia mais Curta (Dijkstra)

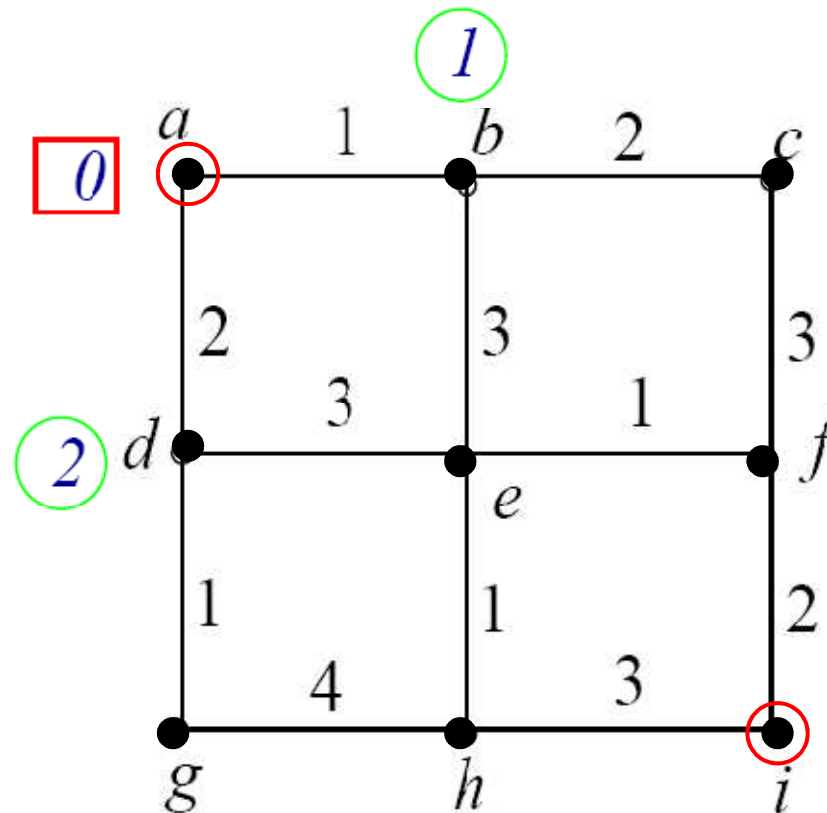
Exemplo 1: Consideremos o seguinte grafo ponderado:



Aplicamos o algoritmo anterior, tomando a como vértice inicial e i como vértice final.

Dado um vértice x , designemos por  uma etiqueta temporária de x e por  a etiqueta definitiva (se as possuir).

Algoritmo da Cadeia mais Curta (Dijkstra)

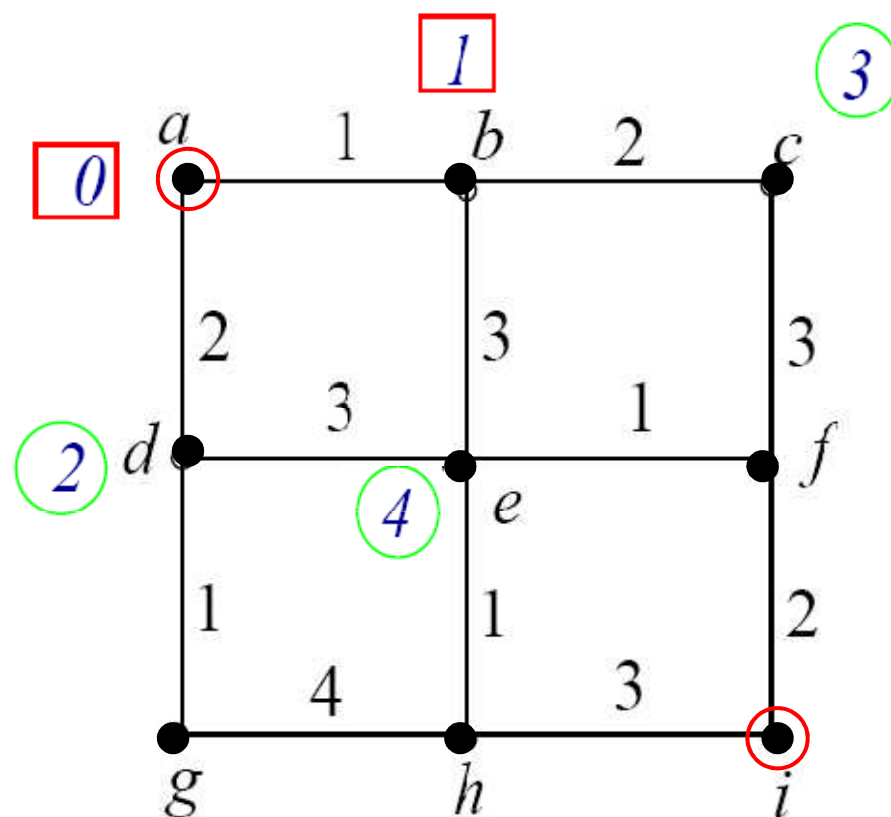


Dado um vértice x , designemos por $\varepsilon_T(x)$ uma etiqueta temporária de x e por $\varepsilon_D(x)$ a etiqueta definitiva (se as possuir).

Então:

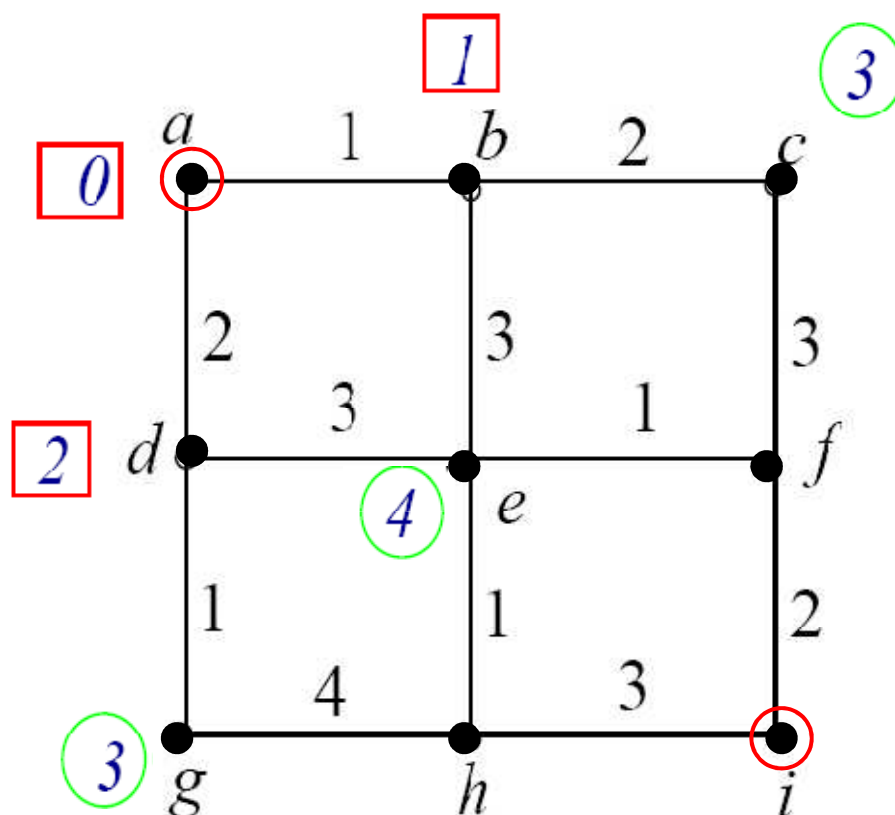
1. Atribuir $\varepsilon_D(a) = 0$, $\varepsilon_T(b) = 1$ e $\varepsilon_T(d) = 2$;

Algoritmo da Cadeia mais Curta (Dijkstra)

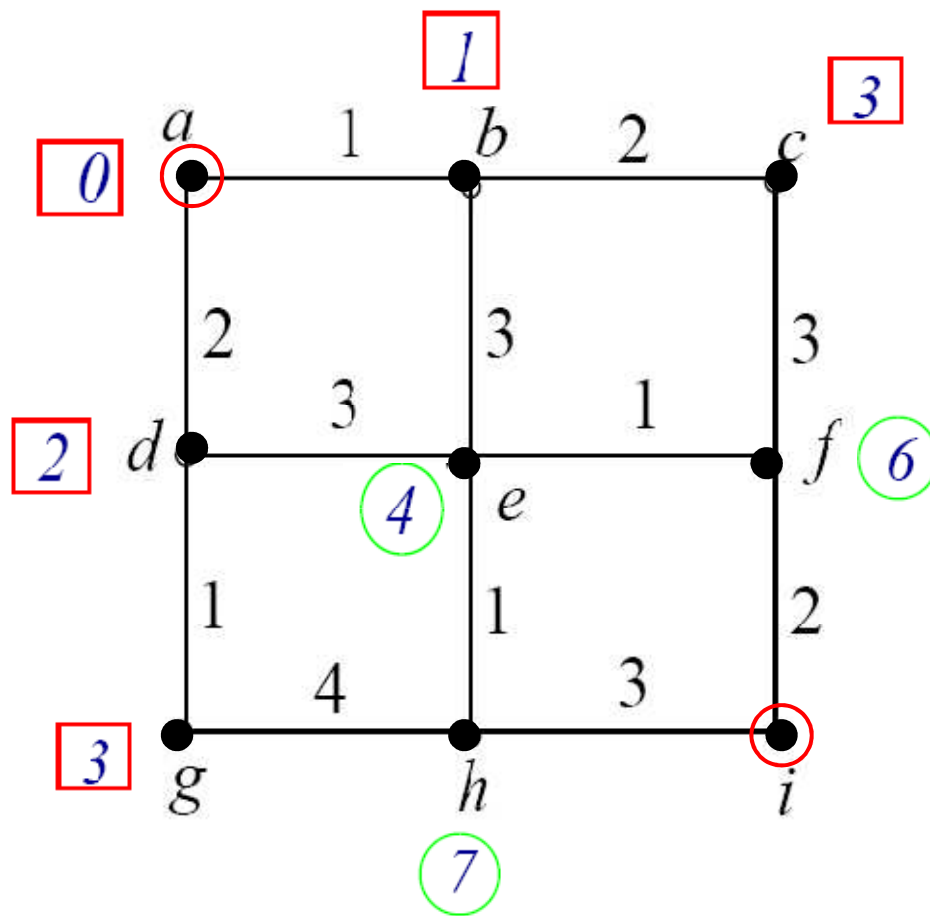


2. Atribuir $\varepsilon_D(b) = 1$, $\varepsilon_T(c) = 1 + 2 = 3$ e $\varepsilon_T(e) = 1 + 3 = 4$;

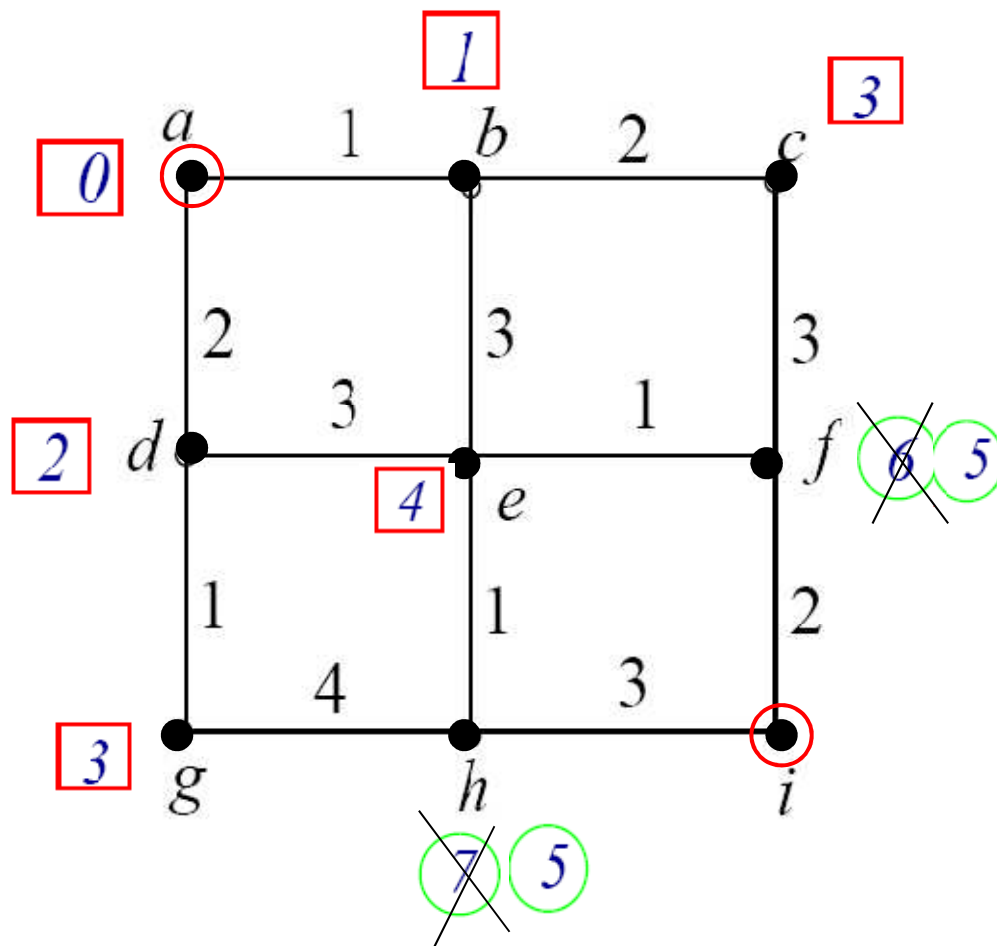
Algoritmo da Cadeia mais Curta (Dijkstra)



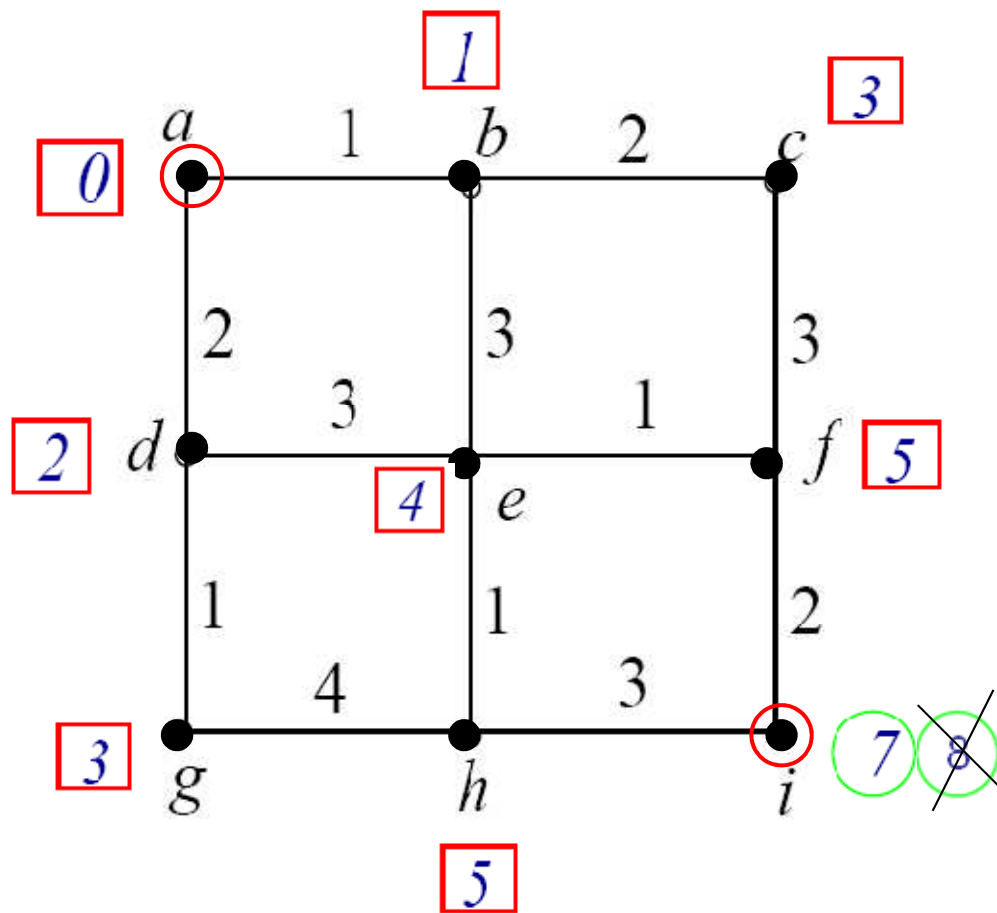
3. Atribuir $\varepsilon_D(d) = 2$, $\varepsilon_T(g) = 2 + 1 = 3$ e manter $\varepsilon_T(e) = 4 < 2 + 3$;



4. Atribuir $\varepsilon_D(c) = 3$, $\varepsilon_D(g) = 3$, $\varepsilon_T(f) = 3 + 3 = 6$ e $\varepsilon_T(h) = 3 + 4 = 7$;

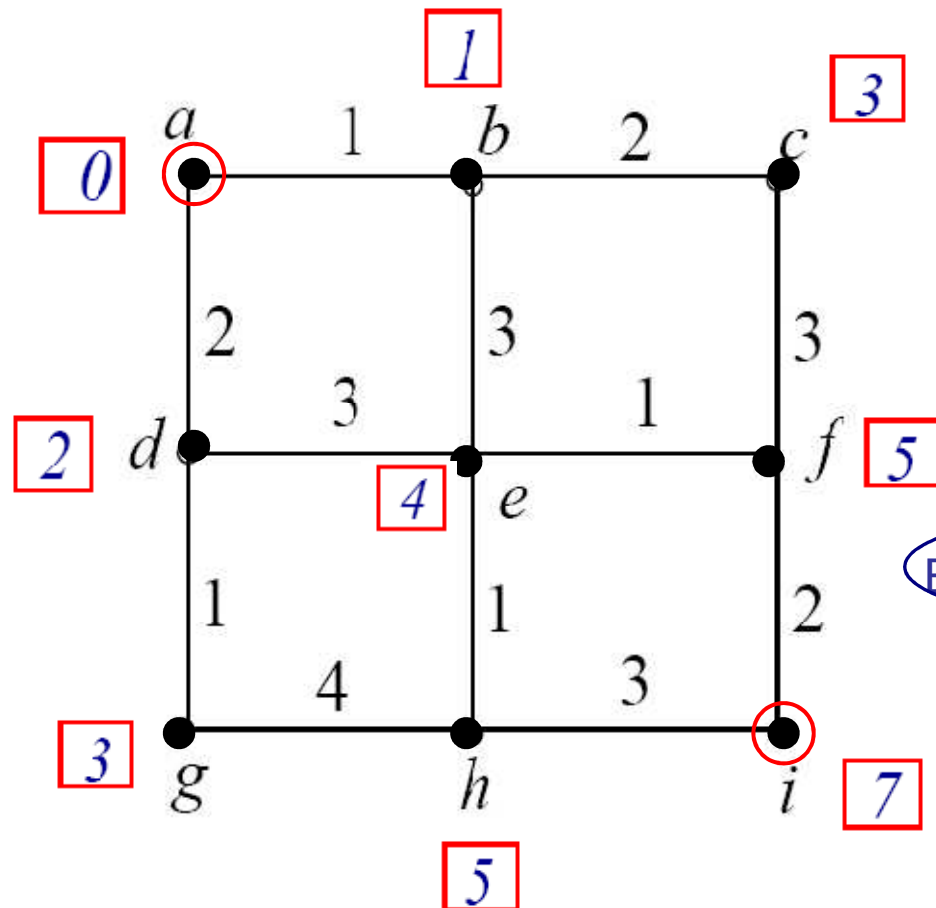


5. Atribuir $\varepsilon_D(e) = 4$, $\varepsilon_T(f) = 4 + 1 = 5 < 6$ e $\varepsilon_T(h) = 4 + 1 = 5 < 7$;



6. Atribuir $\varepsilon_D(f) = 5$, $\varepsilon_D(h) = 5$ e $\varepsilon_T(i) = 5 + 2 = 7 < 5 + 3$;

7. Atribuir $\varepsilon_D(i) = 7$ e terminar.



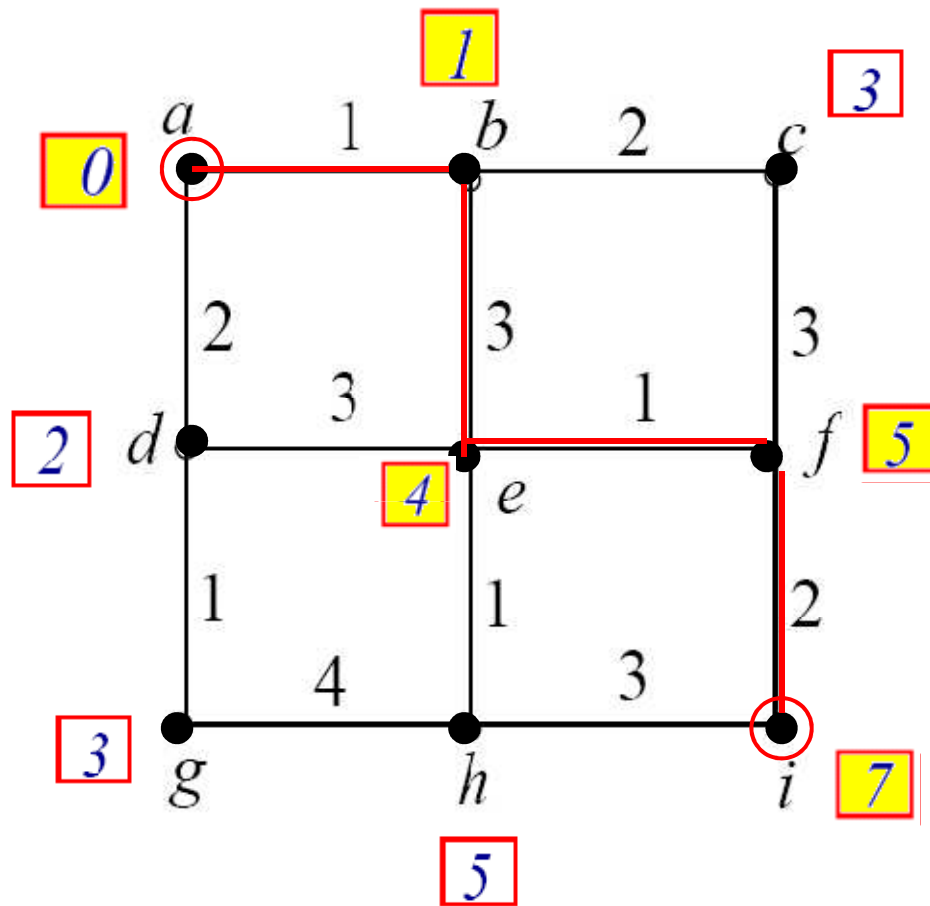
Etiqueta definitiva em i



Valor da etiqueta definitiva de x_i

No final procurar uma sequência de vértices $x_0 = a, x_1, \dots, x_{k-1}, x_k = i$, todos com etiquetas definitivas tais que

$$\{x_{i-1}, x_i\} \in \mathcal{U} \text{ e } \varepsilon(x_i) - \varepsilon(x_{i-1}) = v(\{x_{i-1}, x_i\}).$$



Uma cadeia $a-i$ mínima é $a \rightarrow b \rightarrow e \rightarrow f \rightarrow i$

Proposição 2.3.7:

Sejam (G, v) , com $G = (X, \mathcal{U})$, um grafo conexo ponderado e x e y dois vértices distintos de G .

1. Se $|X| = n$ então o Algoritmo da Cadeia mais Curta, tomando x e y como vértices inicial e final respectivamente, termina ao fim de repetir o PASSO 3 no máximo $n - 2$ vezes;

2. Consideremos as etiquetas dos vértices de G no fim de se aplicar o Algoritmo da Cadeia mais Curta, tomando x e y como vértices inicial e final respectivamente.

Seja z um vértice com etiqueta definitiva ε (em particular, $z = y$) e sejam $x = x_0, x_1, \dots, x_{k-1}, x_k = z$ vértices com etiquetas definitivas

$$\varepsilon(x) = \varepsilon(x_0) = 0, \quad \varepsilon(x_1), \quad \dots, \quad \varepsilon(x_{k-1}), \quad \varepsilon(x_k) = \varepsilon(z) = \varepsilon$$

tais que, para $i \in \{1, \dots, k\}$,

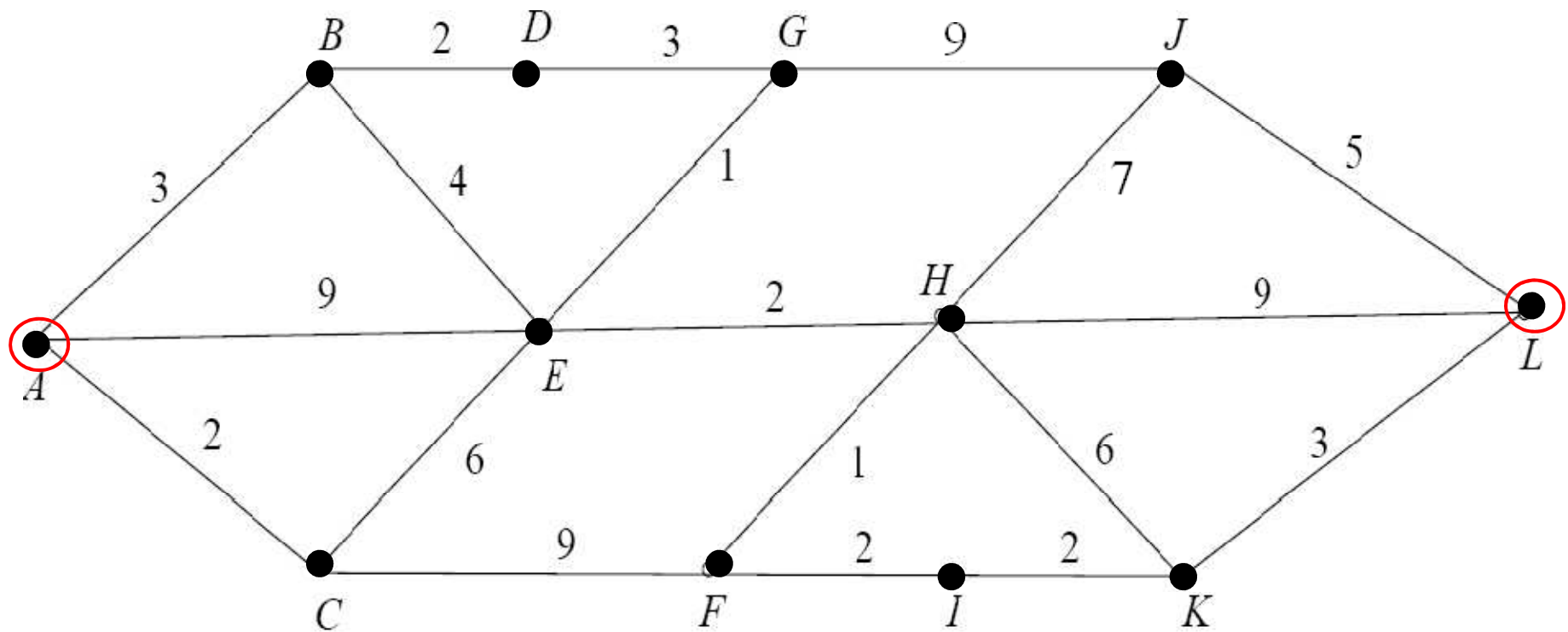
$$\{x_{i-1}, x_i\} \in \mathcal{U} \text{ e } \varepsilon(x_i) - \varepsilon(x_{i-1}) = v(\{x_{i-1}, x_i\}).$$

Então

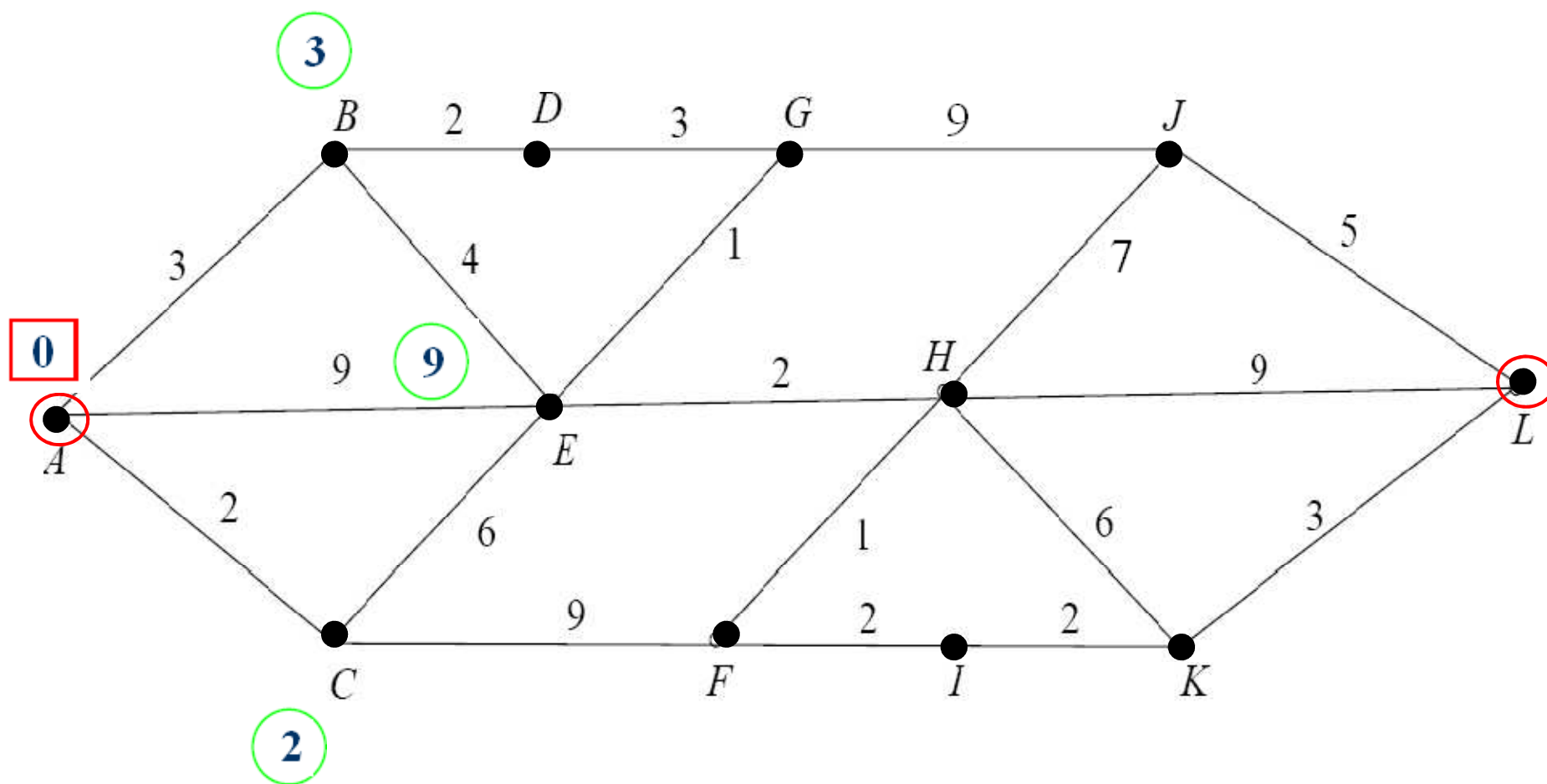
$$x = x_0 \rightarrow x_1 \rightarrow \dots \rightarrow x_{k-1} \rightarrow x_k = z$$

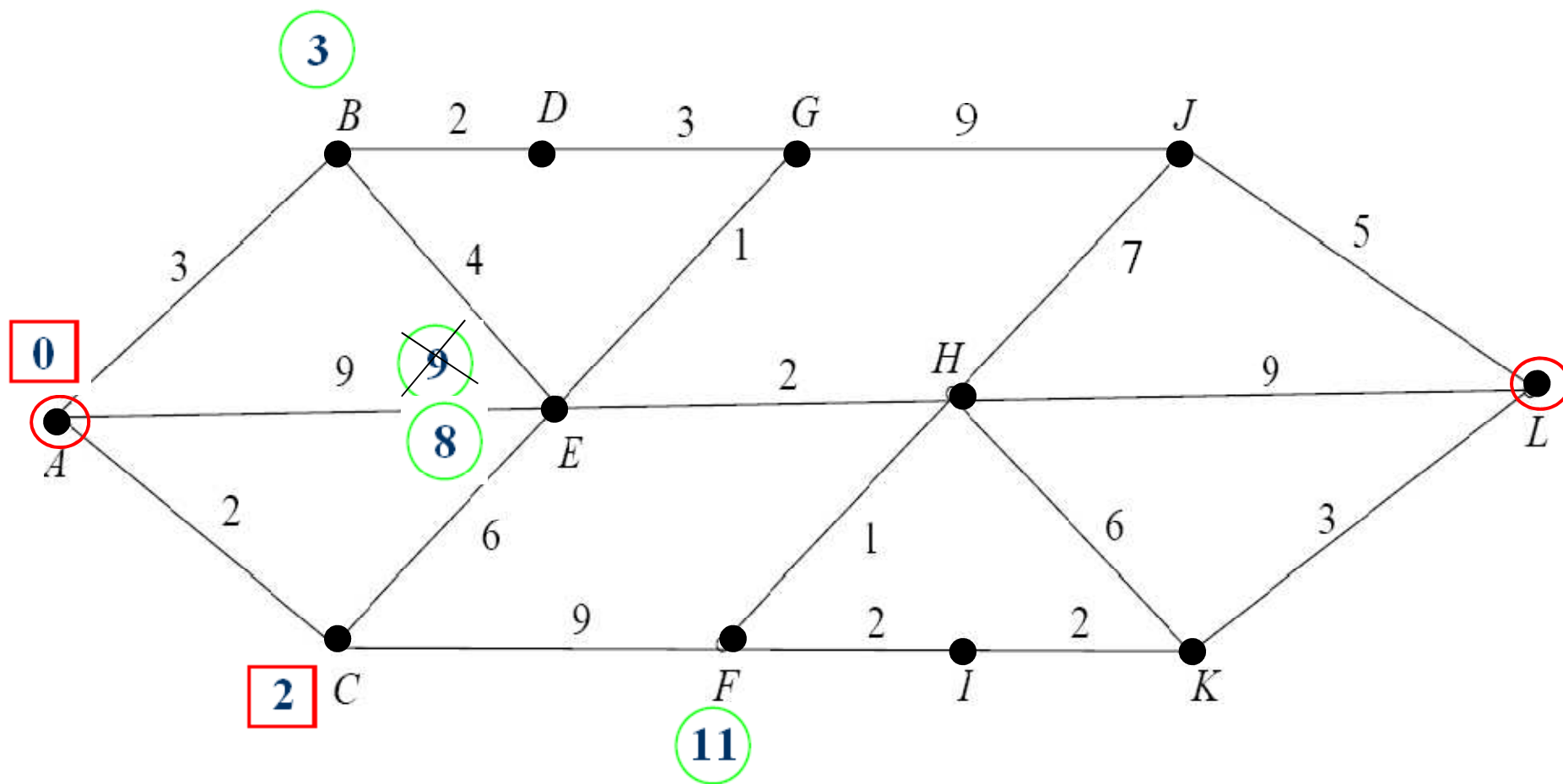
é uma cadeia $x - z$ mínima de valor igual a ε .

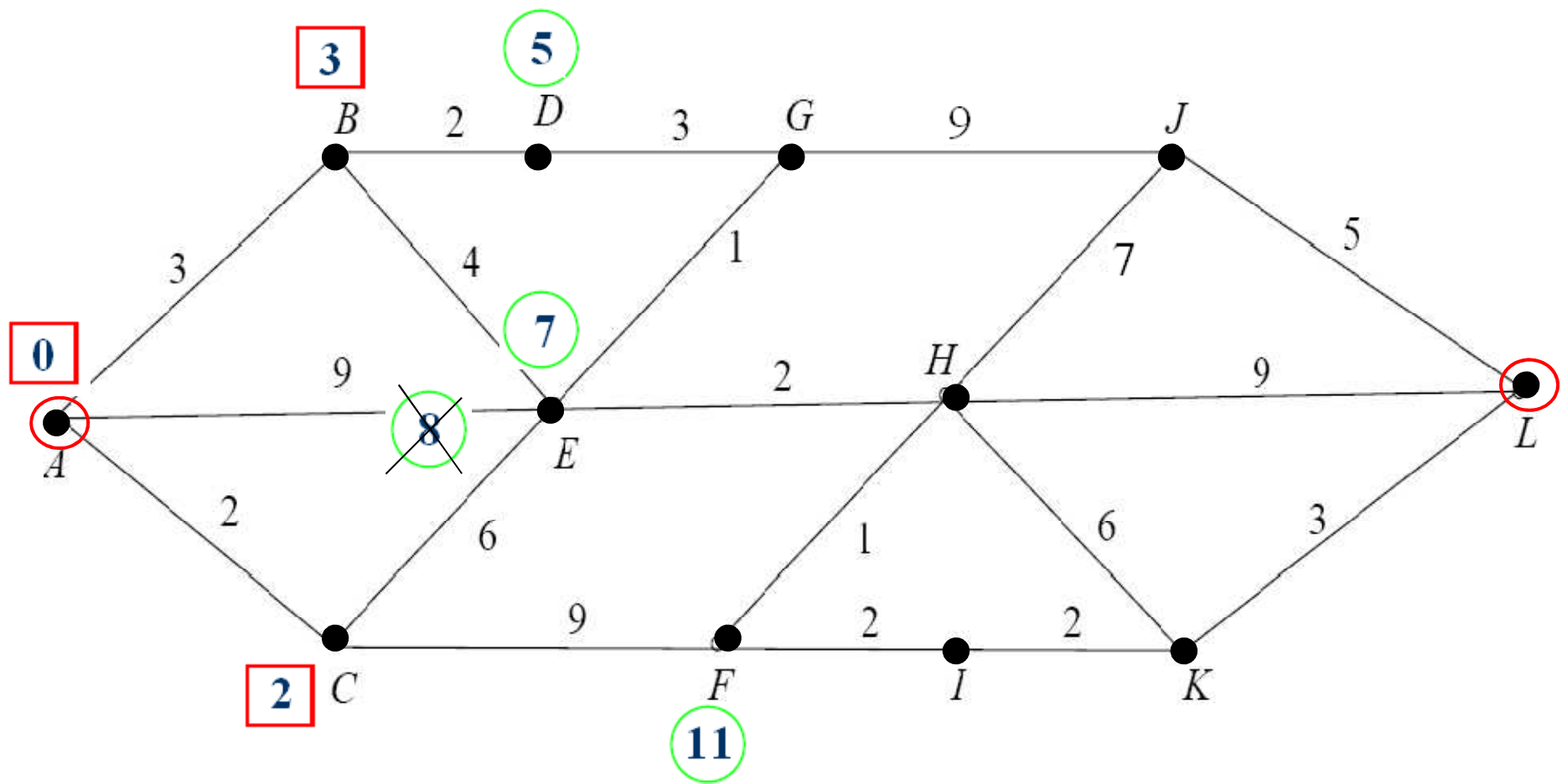
Exemplo 2: Considere o seguinte grafo ponderado:

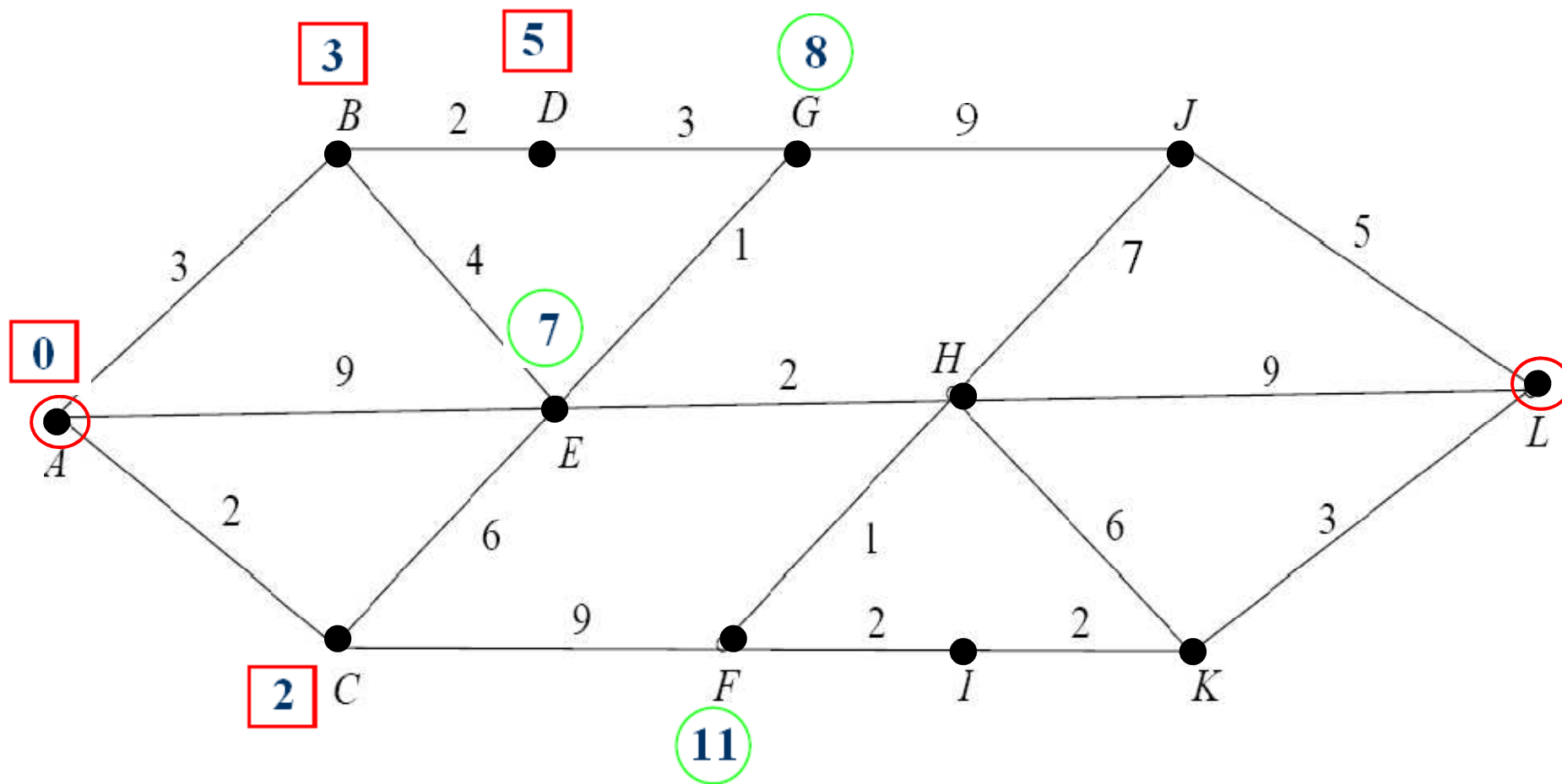


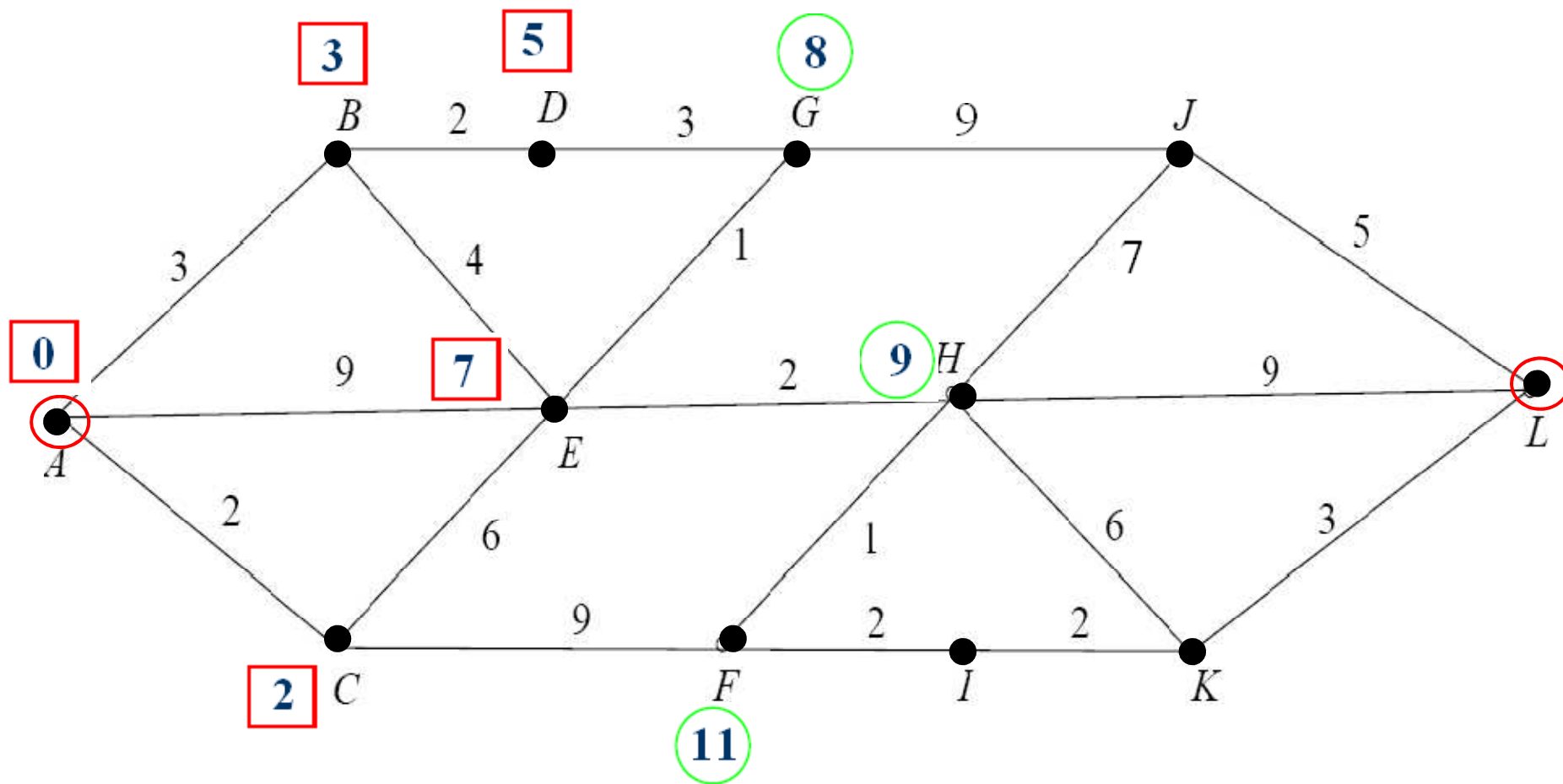
Aplique o Algoritmo da Cadeia mais Curta para obter
uma cadeia $A - L$ mínima

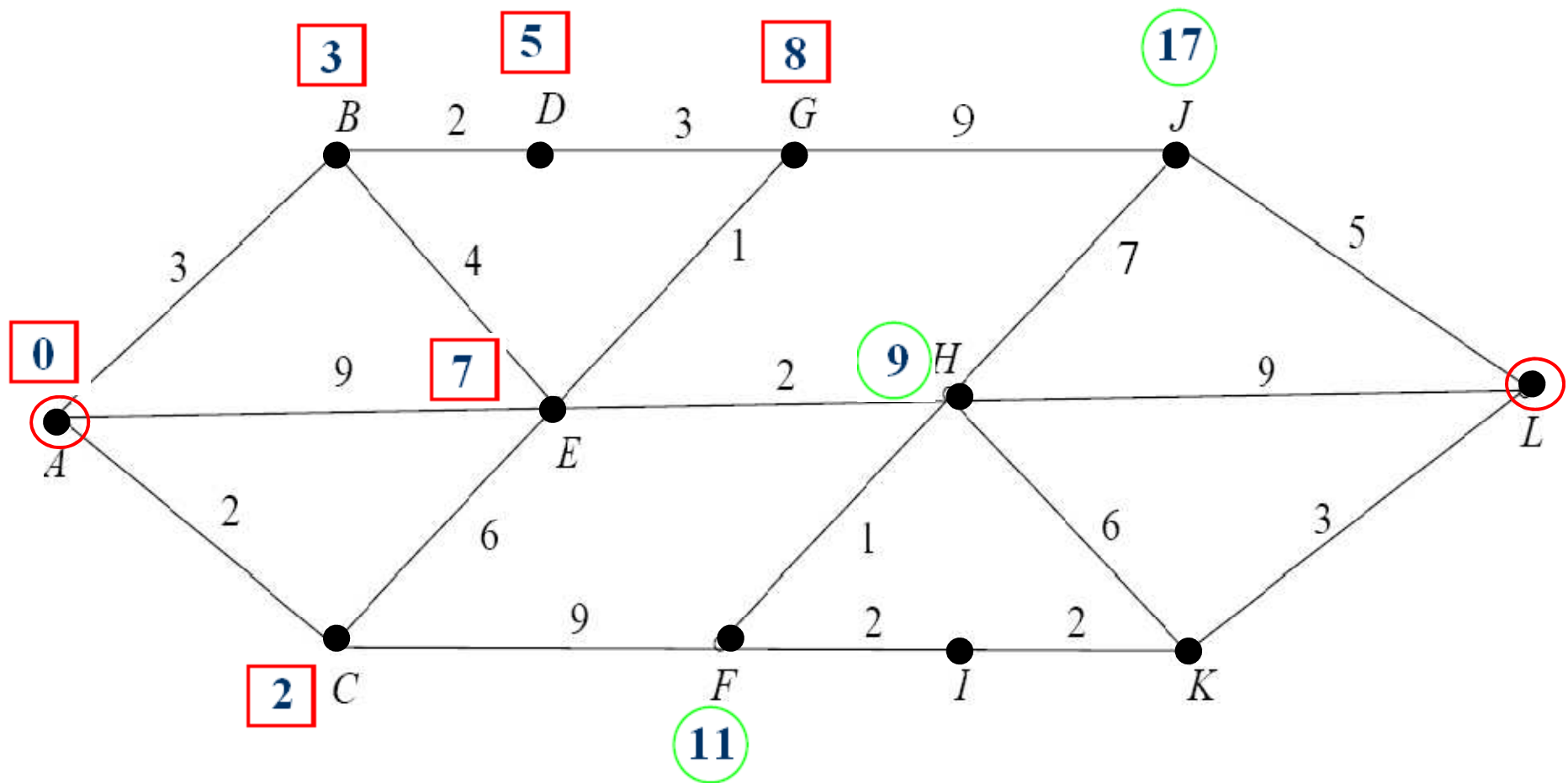


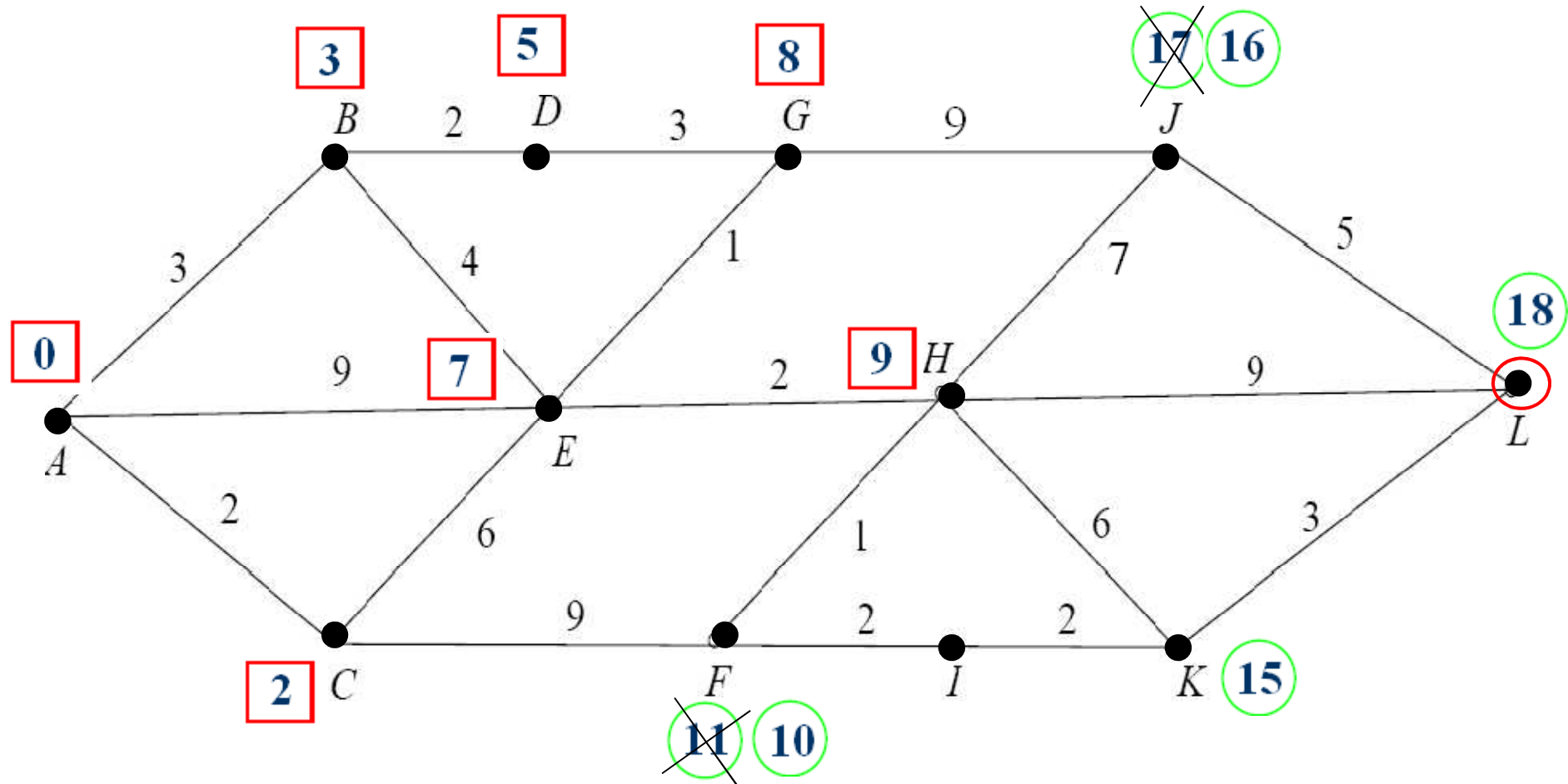


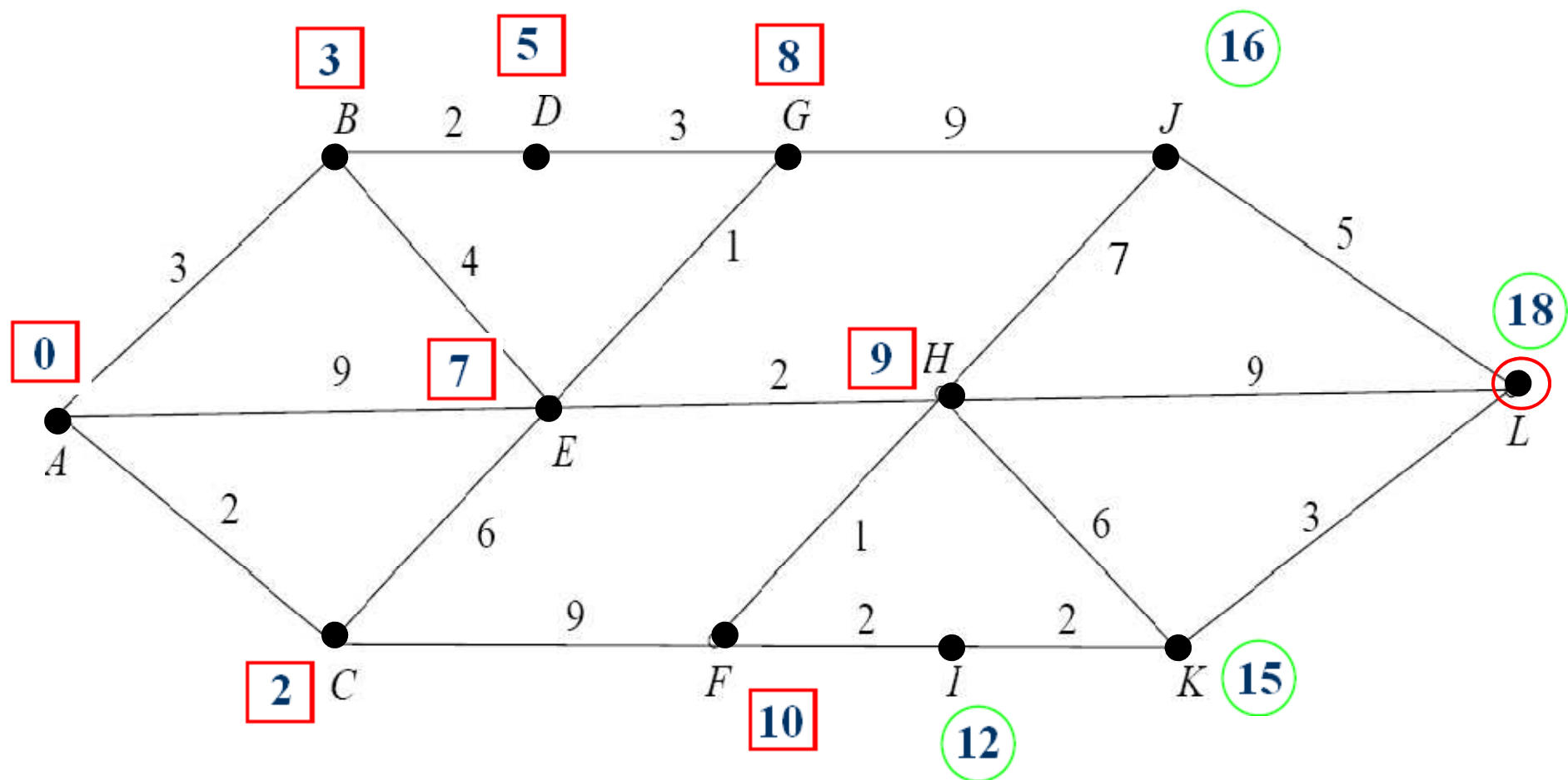


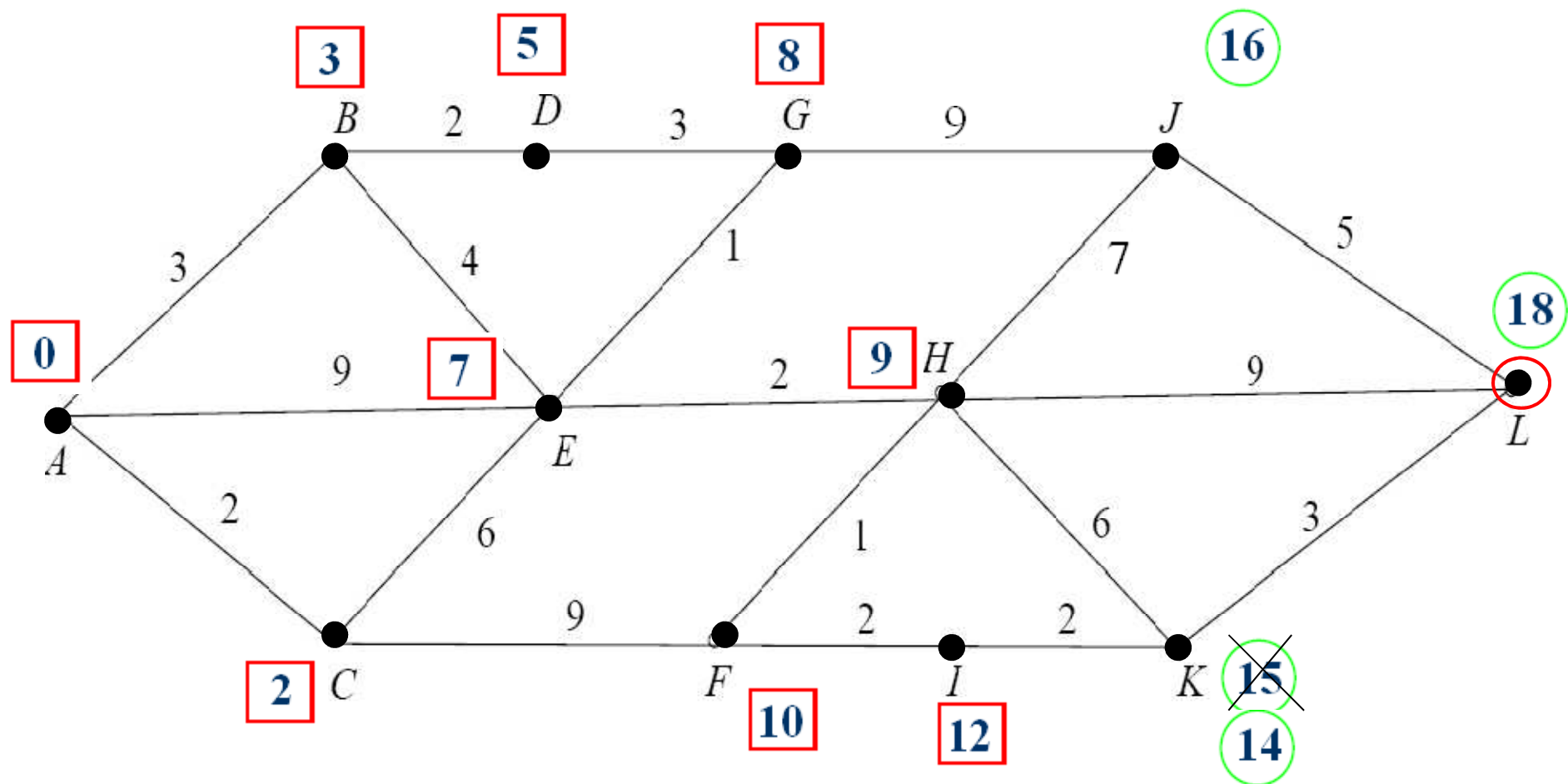


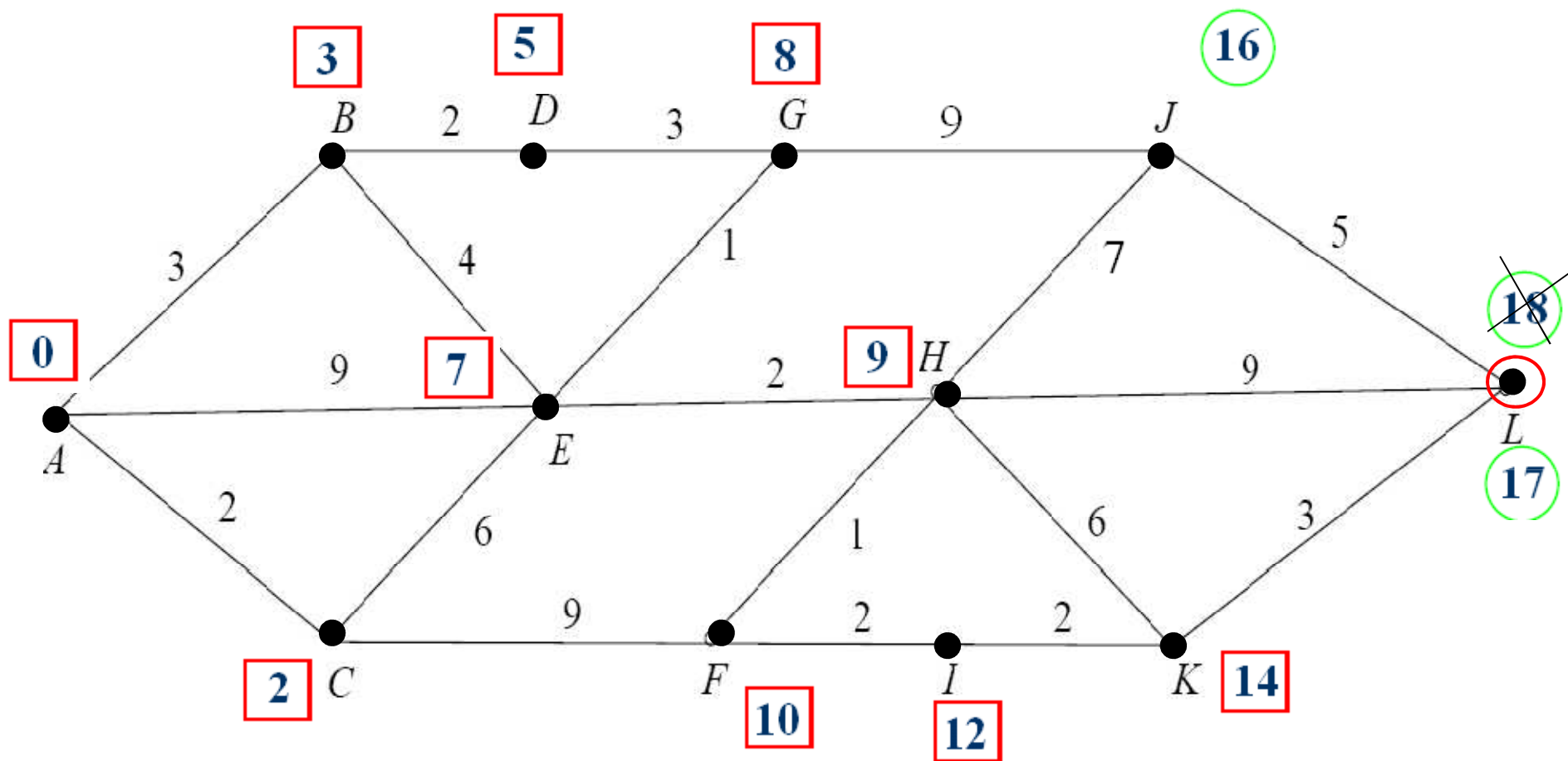


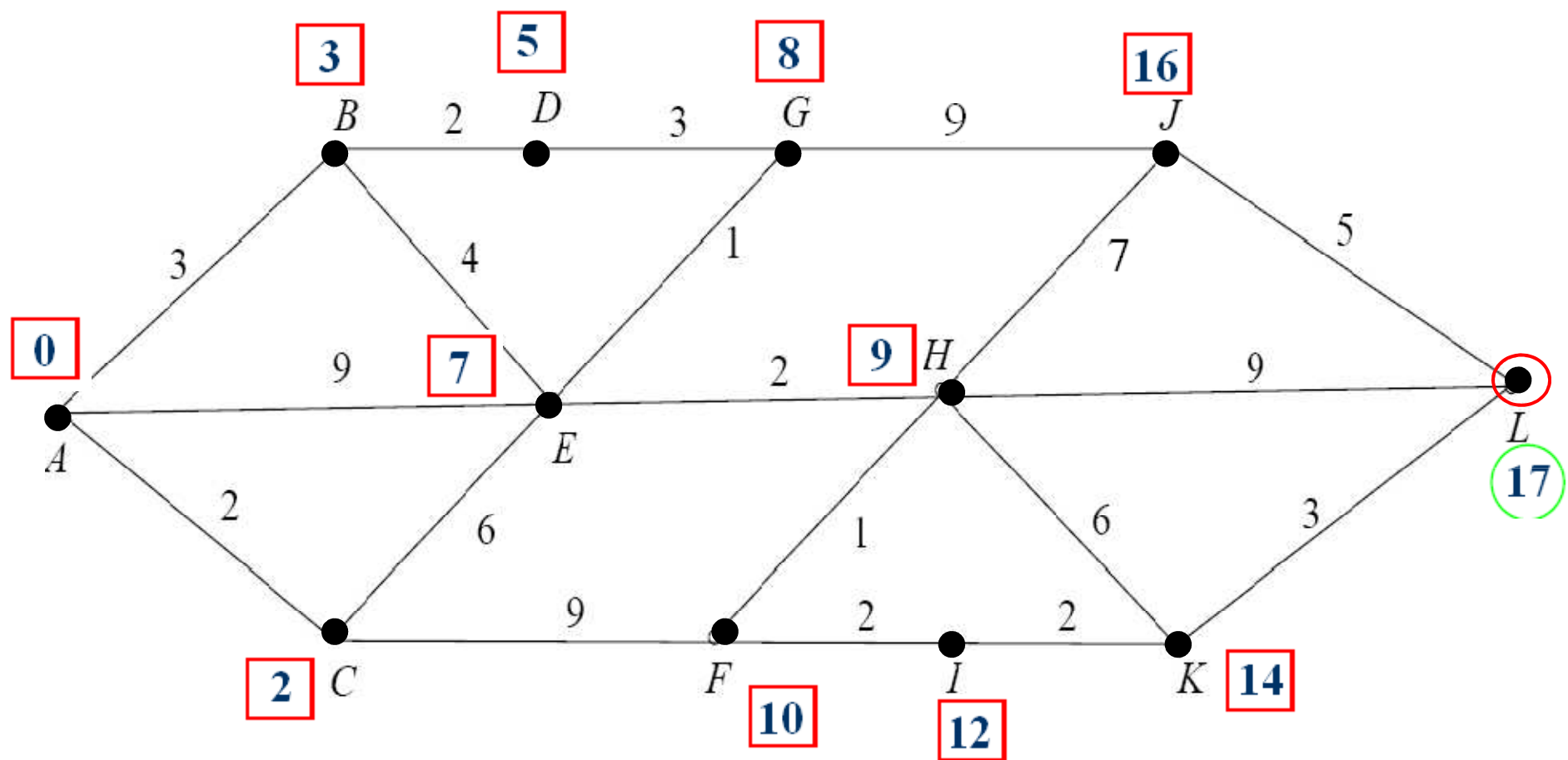


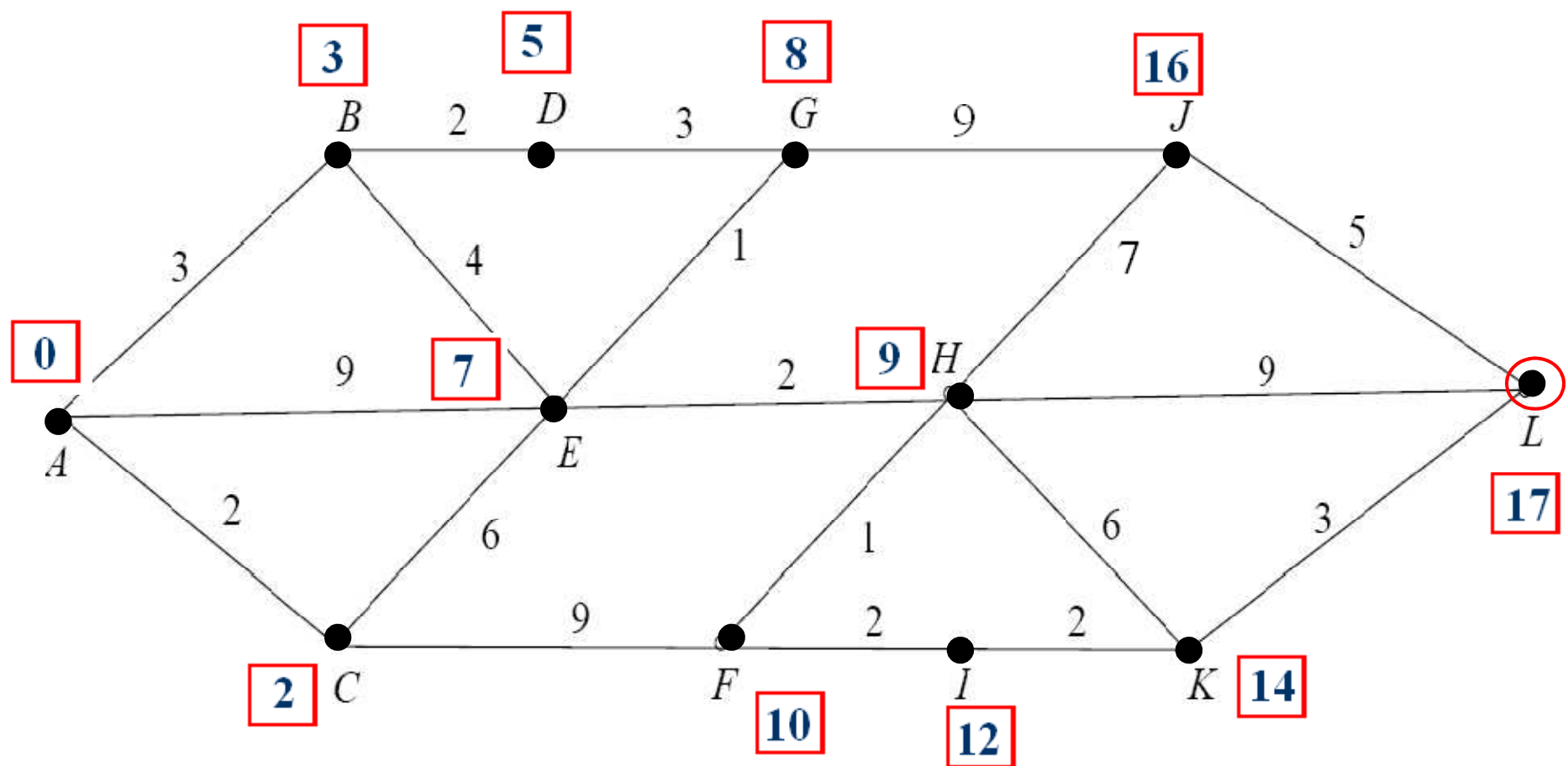




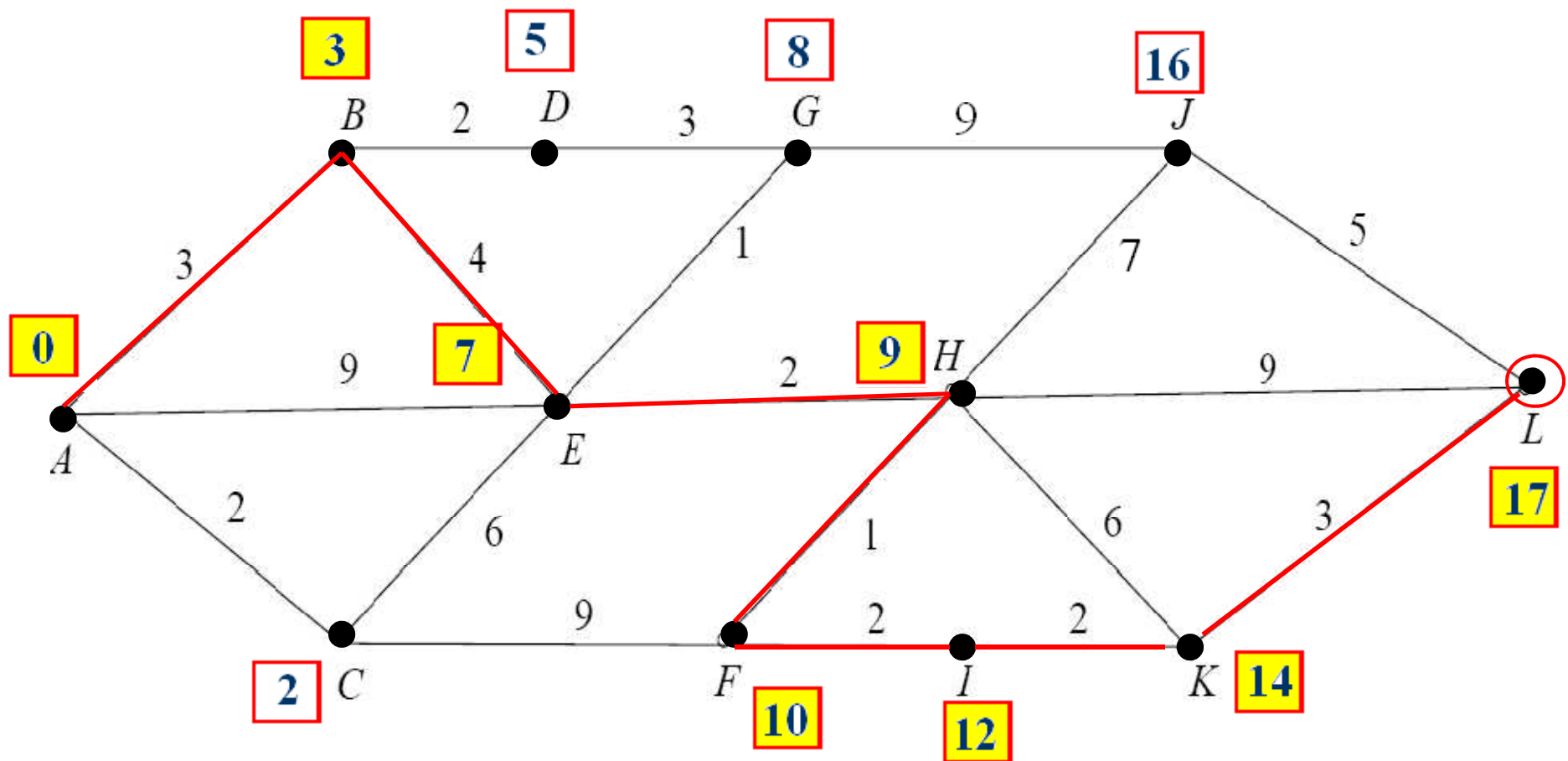








Uma cadeia A-L mínima tem então valor 17.



Uma cadeia A-L mínima de valor 17 é $A \rightarrow B \rightarrow E \rightarrow H \rightarrow F \rightarrow I \rightarrow K \rightarrow L$.