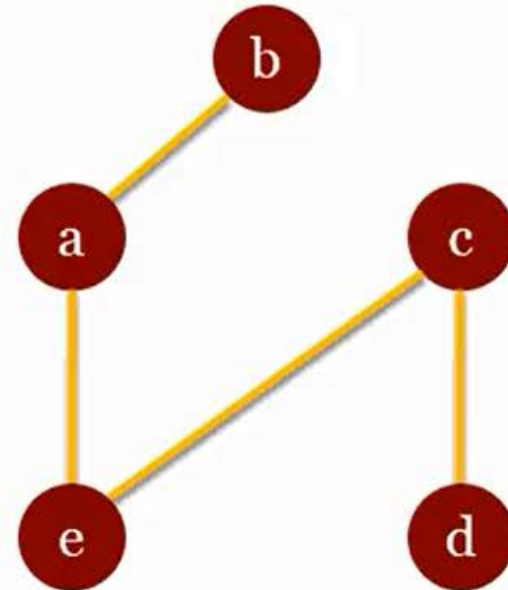
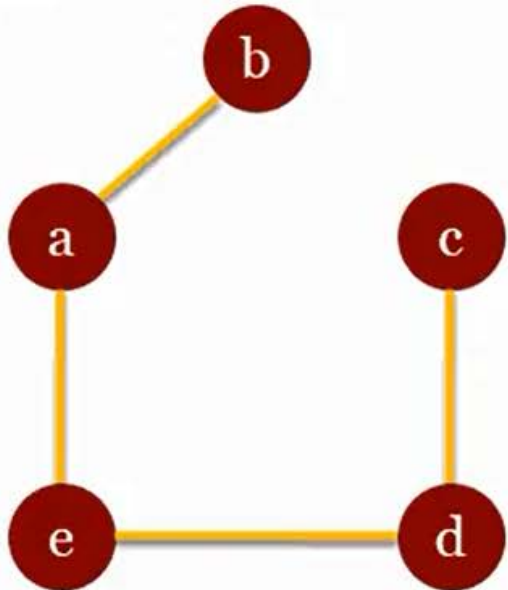
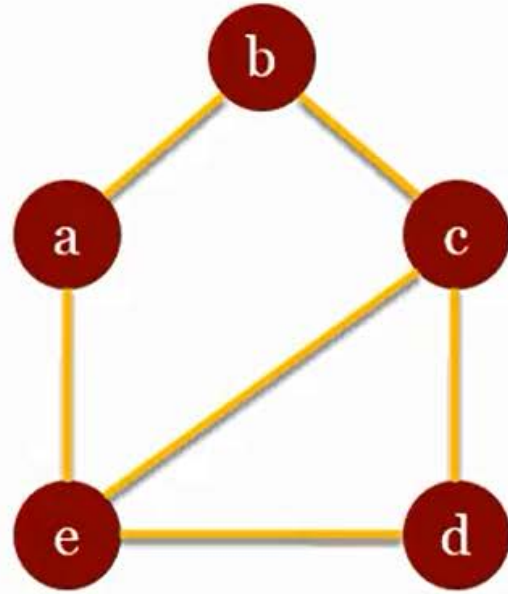


CÂY KHUNG CỦA ĐỒ THỊ

Định nghĩa

- Nếu đồ thị bộ phận T của đồ thị vô hướng liên thông $G = (V, E)$ là một cây thì khi đó cây T được gọi là **cây khung (spanning tree)** của đồ thị G .
- Cây khung còn có tên gọi khác là cây phủ, cây bao trùm, cây tối đại.

Ví dụ về cây khung



Định lý về tính liên thông

- Một đơn đồ thị là liên thông nếu và chỉ nếu nó có cây khung.

Định lý về tính liên thông

- Một đơn đồ thị là liên thông nếu và chỉ nếu nó có cây khung.



Chứng minh:

- Chiều đảo: Giả sử G có cây khung T . Do T là cây nên có đường đi trên T giữa 2 đỉnh bất kỳ (tính chất 5). Vì T là đồ thị bộ phận của G (theo định nghĩa cây khung) nên G có đường đi giữa 2 đỉnh bất kỳ. Do đó, G liên thông

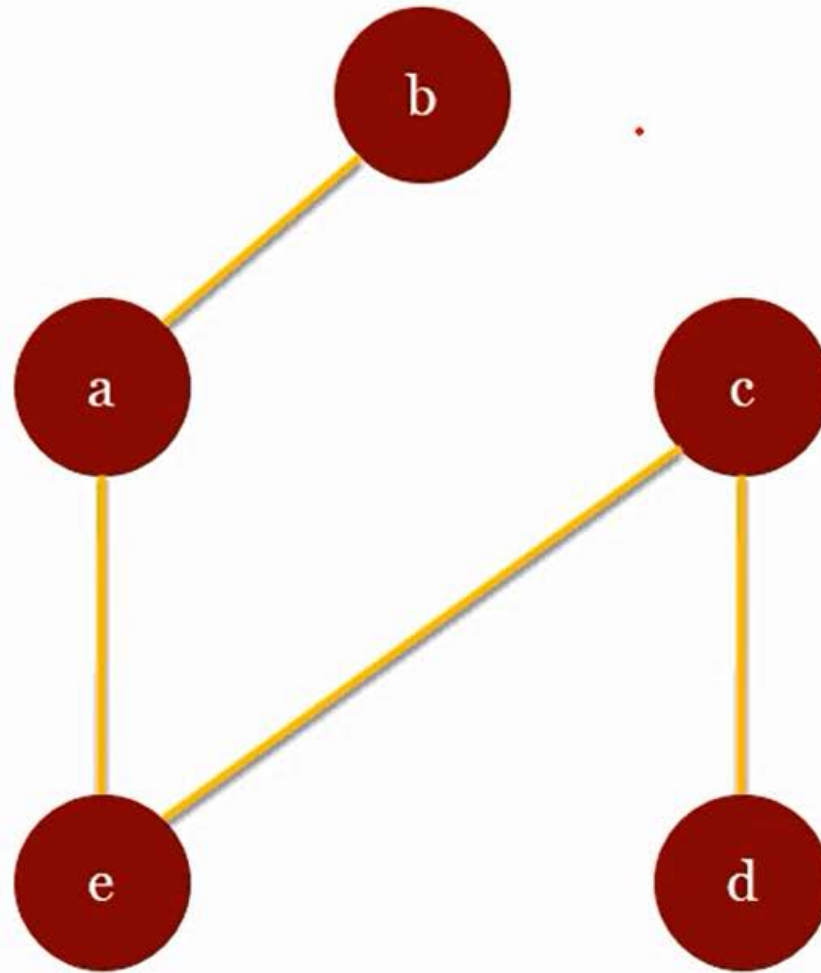
Định lý về tính liên thông

- Chiều thuận: Giả sử G liên thông. Nếu G không phải là cây, thì nó phải có một chu trình đơn. Xóa đi một cạnh bất kỳ trong các chu trình đơn này. Đồ thị nhận được có số cạnh ít hơn đồ thị cũ, nhưng số đỉnh bằng nhau và tính liên thông vẫn còn. Nếu đồ thị bộ phận này không là cây, thì nó vẫn còn chứa chu trình, ta lại lặp lại quá trình xóa cạnh cho đến khi không còn chu trình đơn. Điều này có thể vì số cạnh là hữu hạn. Quá trình kết thúc khi không còn chu trình đơn trong đồ thị. Cây được tạo ra vẫn còn liên thông. Cây này, hơn nữa là cây khung vì nó chứa toàn bộ các đỉnh của G .

Xác định cây khung

- Ta có thể sử dụng 2 thuật toán tìm kiếm theo chiều rộng và theo chiều sâu trong chương Đại cương Lý thuyết đồ thị để xác định cây khung.
- Quá trình duyệt các đỉnh chính là quá trình tìm cây khung của đồ thị.

Ví dụ về tìm cây khung



Định lý về số cây khung

- Định lý Borchart, Sylvester, Cayley:

Số cây khung của đồ thị đầy đủ K_n là n^{n-2}

Bài tập

1. Giả sử đồ thị G liên thông, có 13 đỉnh và 20 cạnh. Cây bao trùm của G có bao nhiêu đỉnh? Có bao nhiêu cạnh?

2. Đồ thị đầy đủ K_5 có bao nhiêu cây khung?

*

3. Tìm cây khung của đồ thị sau lần lượt bằng DFS rồi BFS. Chọn đỉnh đầu tiên làm gốc:

	1	2	3	4	5
1		1			1
2	1		1		
3		1		1	
4			1		1
5	1			1	

4. Tìm cây khung của đồ thị sau lần lượt bằng DFS rồi BFS. Chọn đỉnh đầu tiên làm gốc:

	1	2	3	4	5	6	7	8	9	10
1		1			1	1				
2	1		1				1			
3		1		1				1		
4			1		1				1	
5	1			1		1				1
6	1							1	1	
7		1							1	1
8			1			1				
9				1		1	1			
10					1		1	1		

CÂY KHUNG NHỎ NHẤT & CÂY KHUNG LỚN NHẤT

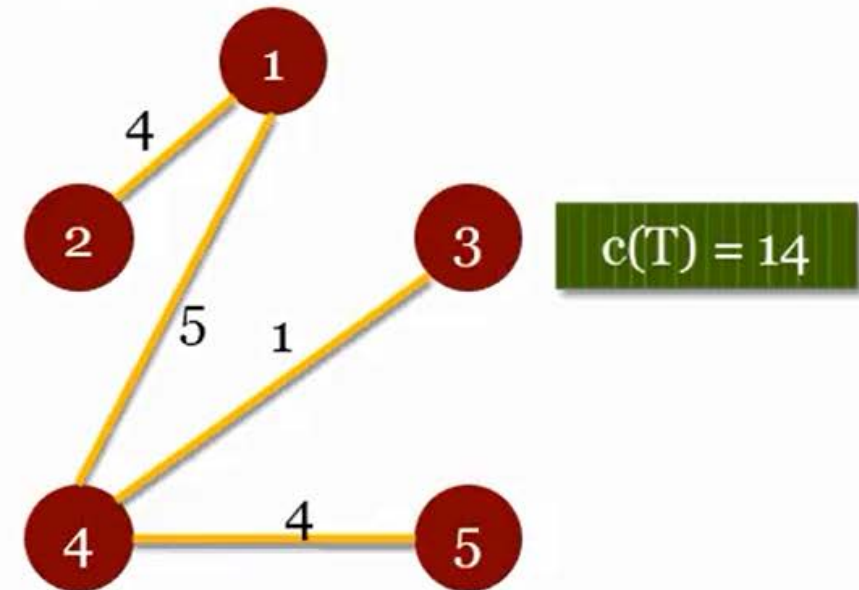
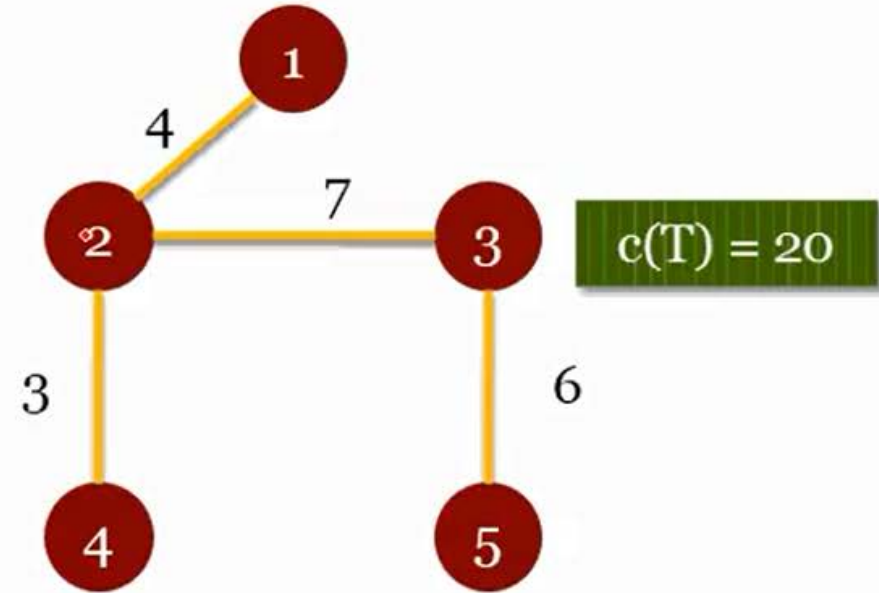
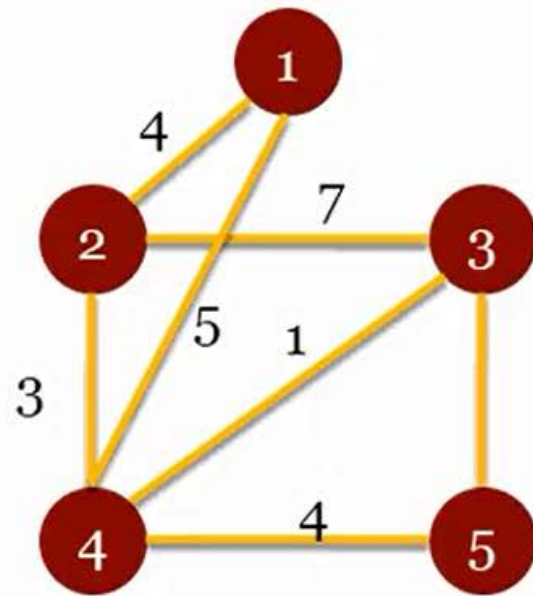
Độ dài của cây khung

- Cho $G = (V, E)$ là đồ thị vô hướng, liên thông. Mỗi cạnh $e \in E$ của đồ thị được gán một **trọng số (weight)** hay **chi phí (cost)** không âm $c(e)$, gọi là **độ dài (length)** của cạnh đó. Giả sử $T = (V_T, E_T)$ là cây khung của đồ thị G . Ta gọi độ dài $c(T)$ của cây khung T là tổng độ dài các cạnh của nó:

$$c(T) = \sum_{e \in E_T} c(e)$$

- Một đồ thị mà các cạnh được gán trọng số như trên, được gọi là **đồ thị có trọng số (weighted graph)**.

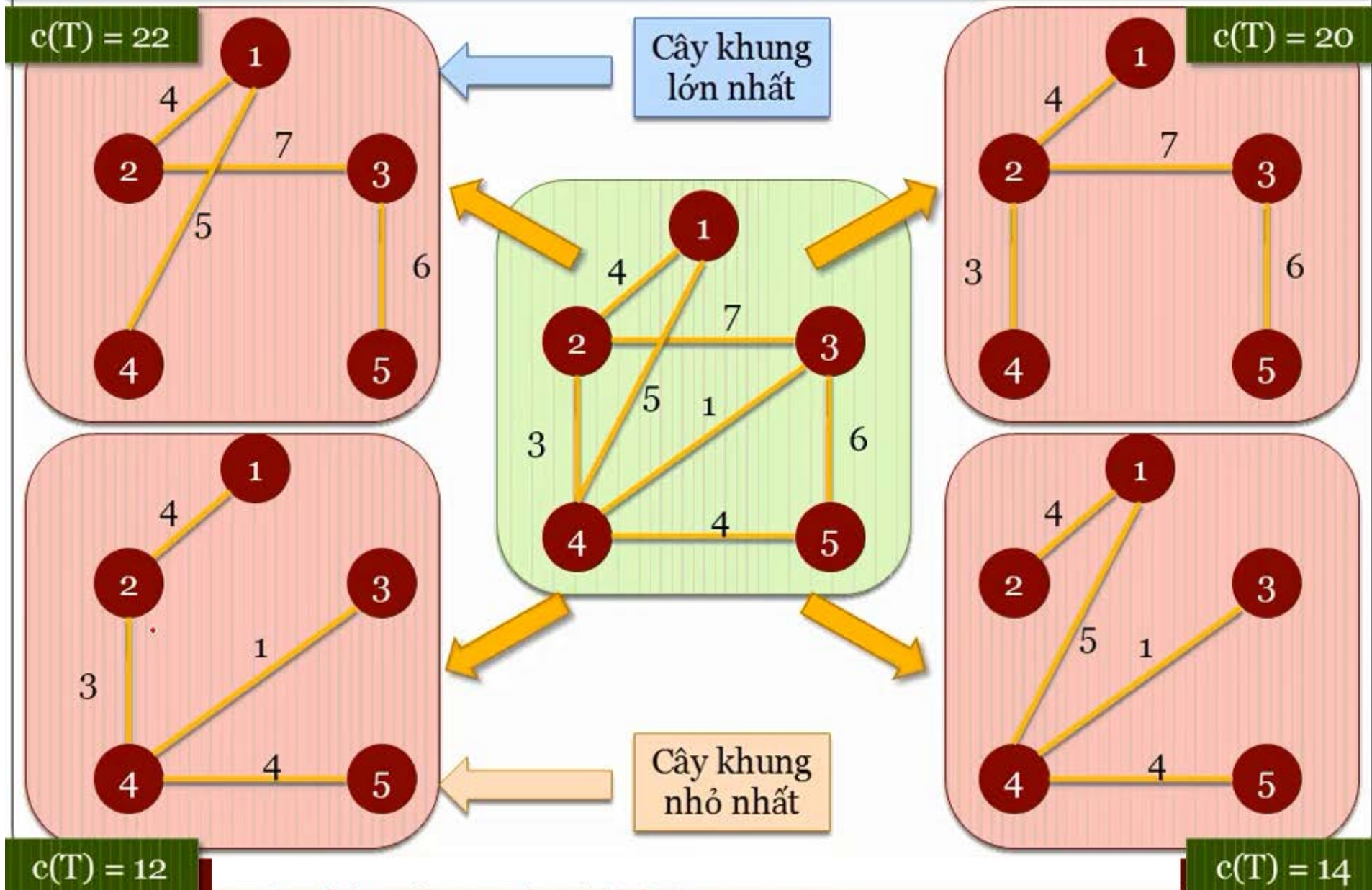
Ví dụ về độ dài cây khung



Phát biểu bài toán

- Trong số tất cả các cây khung của đồ thị G , hãy tìm cây khung với độ dài nhỏ nhất. Cây khung như vậy được gọi là **cây khung nhỏ nhất** (minimum spanning tree) của đồ thị và bài toán đặt ra được gọi là bài toán cây khung nhỏ nhất.
- Tương tự như vậy, cây khung có độ dài lớn nhất được gọi là cây khung lớn nhất.

Ví dụ



Một số bài toán ứng dụng

- **Bài toán xây dựng hệ thống đường cao tốc:** Giả sử ta muốn xây dựng một hệ thống đường cao tốc nối n thành phố sao cho hành khách có thể đi từ một thành phố bất kỳ đến các thành phố còn lại. Mặt khác, trên quan điểm kinh tế đòi hỏi là chi phí về xây dựng hệ thống đường phải là nhỏ nhất.
- Rõ ràng, đây là đồ thị mà đỉnh là các thành phố còn cạnh ứng với các tuyến đường và bài toán đặt ra là phải tìm cây khung nhỏ nhất trong đó trọng số là chi phí thực hiện các tuyến đường cao tốc.

Một số bài toán ứng dụng

- **Bài toán nối mạng máy tính:** Cần nối mạng một hệ thống gồm n máy tính đánh số từ 1 đến n . Biết chi phí nối máy i với máy j là $c[i, j]$. Hãy tìm cách nối mạng sao cho tổng chi phí nối mạng là nhỏ nhất.

Thuật toán tìm cây khung nhỏ nhất

- Prim và Kruskal là hai thuật toán thông dụng để tìm cây khung nhỏ nhất.
 - Thuật toán Prim do *Robert Prim* đưa ra vào năm 1957
 - Thuật toán Kruskal do *Joseph Kruskal* phát minh năm 1956.



Kruskal (1928 -)



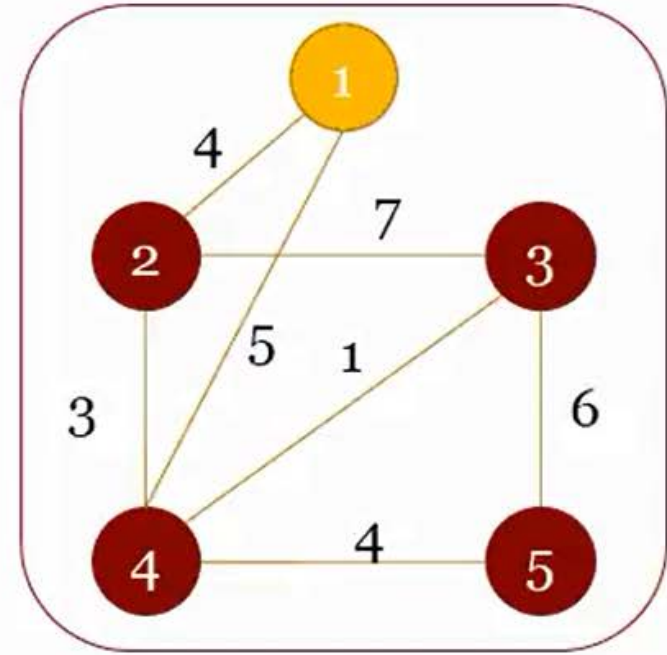
Prim (1921 -)

Thuật toán Prim

- Cho $G = (V, E)$ là một đồ thị liên thông có trọng số gồm n đỉnh.
- **Bước 1.** Chọn tùy ý một đỉnh bất kỳ $v \in V$ và khởi tạo: $Y = \{v\}$ và $T = \emptyset$.
- **Bước 2.** Trong số những cạnh $e = (v, w)$, trong đó $v \in Y$ và $w \in V \setminus Y$, ta chọn cạnh có độ dài nhỏ nhất.
- **Bước 3.** Gán $Y = Y \cup \{w\}$ và $T = T \cup \{e\}$
- **Bước 4.** Nếu T đủ $n - 1$ phần tử thì dừng, ngược lại làm tiếp bước 2.
- *T chính là cây khung nhỏ nhất.*

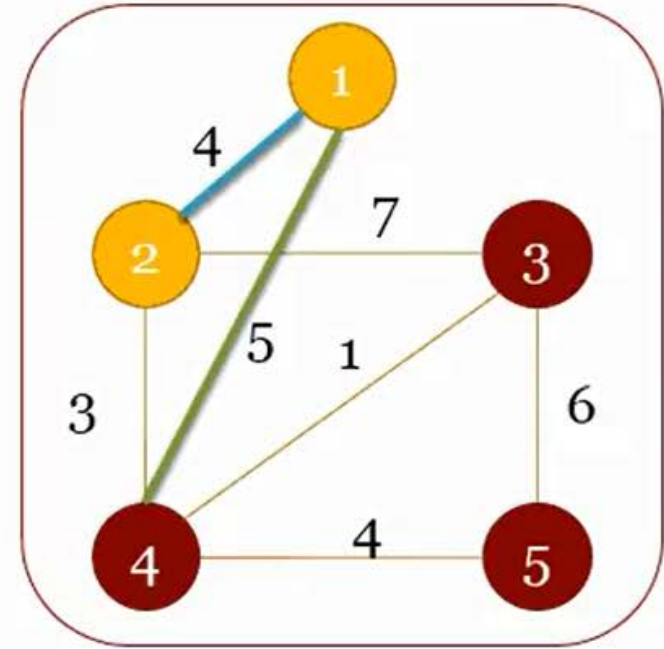
Ví dụ thuật toán Prim

- Giả sử chọn đỉnh 1 là đỉnh đầu
- Cập nhật
 - $Y = \{1\}$
 - $T = \emptyset$



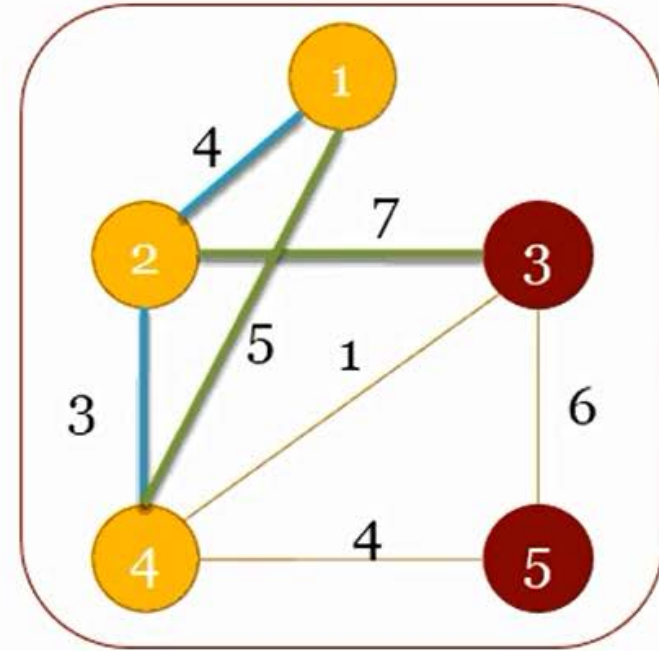
Ví dụ thuật toán Prim

- Chọn cạnh (1, 2) vì là cạnh có trọng nhỏ nhất trong số các cạnh thỏa yêu cầu ở bước 2: (1, 2), (1, 4)
- Cập nhật:
 - $Y = \{1, 2\}$
 - $T = \{(1, 2)\}$



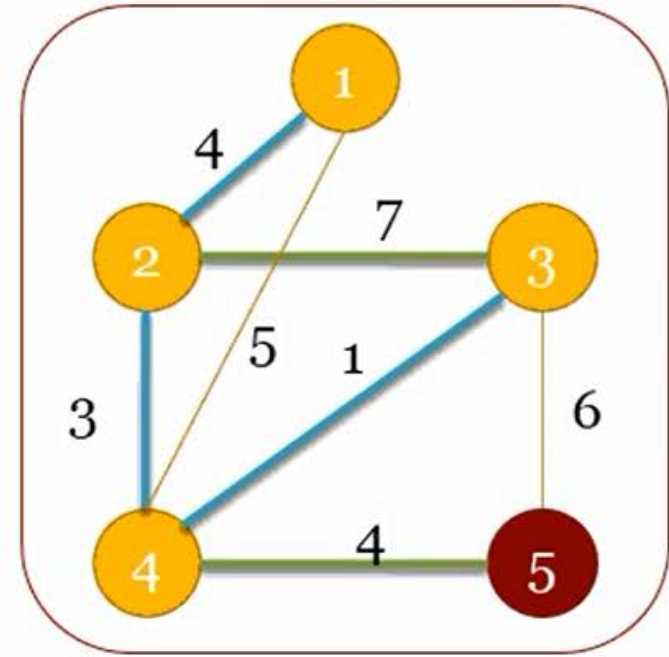
Ví dụ thuật toán Prim

- Chọn cạnh (2, 4) vì là cạnh có trọng nhỏ nhất trong số các cạnh thỏa yêu cầu ở bước 2: (1, 4), (2, 3), (2, 4)
- Cập nhật:
 - $Y = \{1, 2, 4\}$
 - $T = \{\{1, 2\}, \{2, 4\}\}$



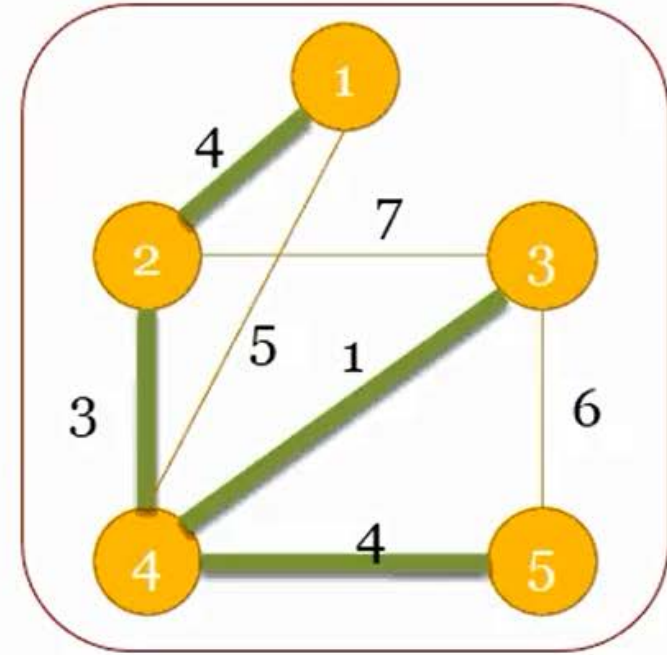
Ví dụ thuật toán Prim

- Chọn cạnh (4, 3) vì là cạnh có trọng nhỏ nhất trong số các cạnh thỏa yêu cầu ở bước 2: (2, 3), (4, 3), (4, 5)
- Cập nhật:
 - $Y = \{1, 2, 4, 3\}$
 - $T = \{\{1, 2\}, \{2, 4\}, \{4, 3\}\}$



Ví dụ thuật toán Prim

- Kết thúc thuật toán vì $|T| = 4$.
- Cây khung nhỏ nhất tìm được có trọng bằng 12.
 - $Y = \{1, 2, 3, 4, 5\}$
 - $T = \{\{1, 2\}, \{2, 4\}, \{3, 4\}, \{4, 5\}\}$

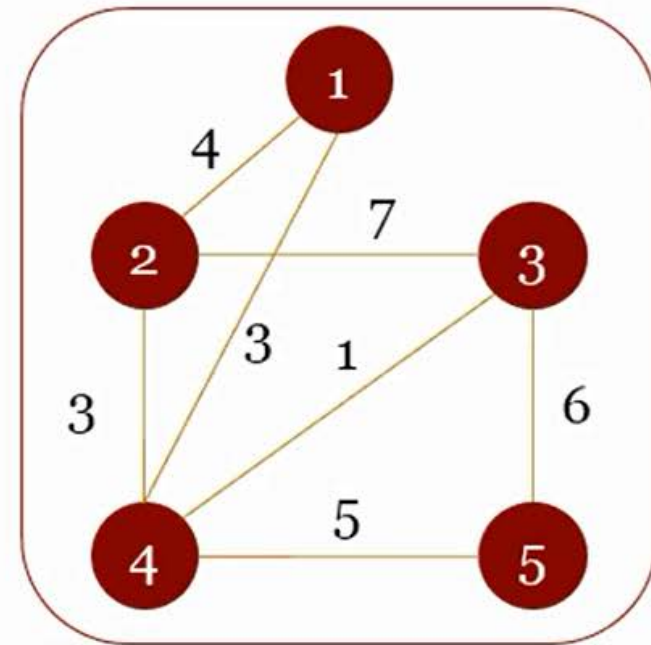


Thuật toán Kruskal

- Cho $G = (V, E)$ là một đồ thị liên thông có trọng số gồm n đỉnh.
- **Bước 1.** Sắp xếp các cạnh theo thứ tự độ dài tăng dần và khởi tạo: $T = \emptyset$.
- **Bước 2.** Lần lượt lấy từng cạnh e trong danh sách đã sắp xếp. Nếu $T \cup \{e\}$ không tạo thành chu trình thì gán $T = T \cup \{e\}$
- **Bước 3.** Nếu T đủ $n - 1$ phần tử thì dừng, ngược lại làm tiếp bước 2.
- *T chính là cây khung nhỏ nhất.*

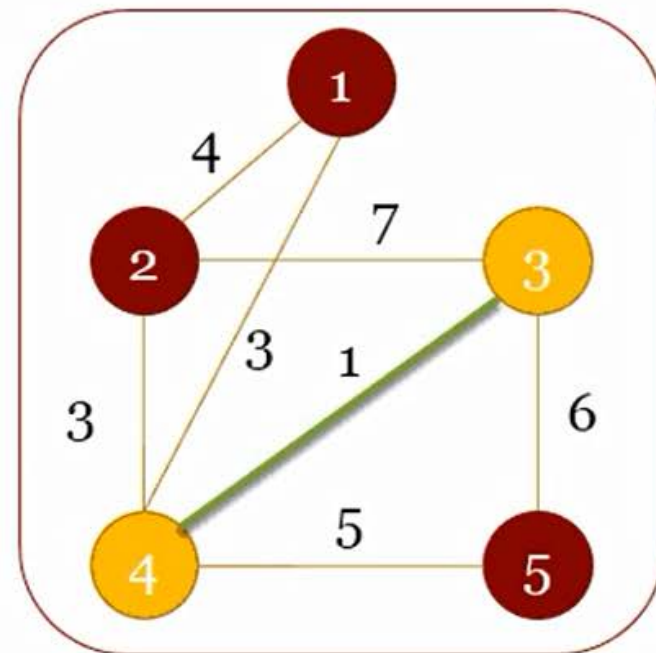
Ví dụ thuật toán Kruskal

- Sắp xếp các cạnh theo thứ tự tăng dần:
 - $(3, 4) = 1$
 - $(1, 4) = 3$
 - $(2, 4) = 3$
 - $(1, 2) = 4$
 - $(4, 5) = 5$
 - $(3, 5) = 6$
 - $(2, 3) = 7$
- $T = \emptyset$



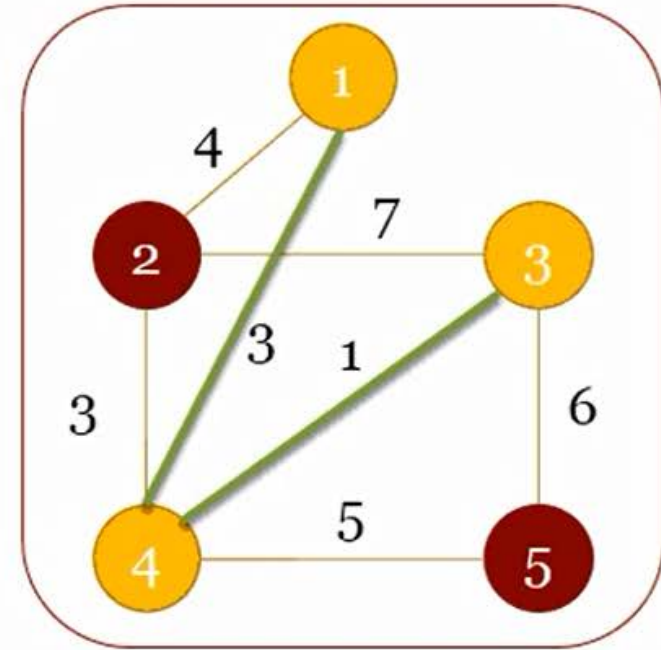
Ví dụ thuật toán Kruskal

- Chọn cạnh (3, 4) – là cạnh có trọng nhỏ nhất.
- $T = \{(3, 4)\}$
- Danh sách các cạnh tăng dần:
 - (3, 4) = 1
 - (1, 4) = 3
 - (2, 4) = 3
 - (1, 2) = 4
 - (4, 5) = 5
 - (3, 5) = 6
 - (2, 3) = 7



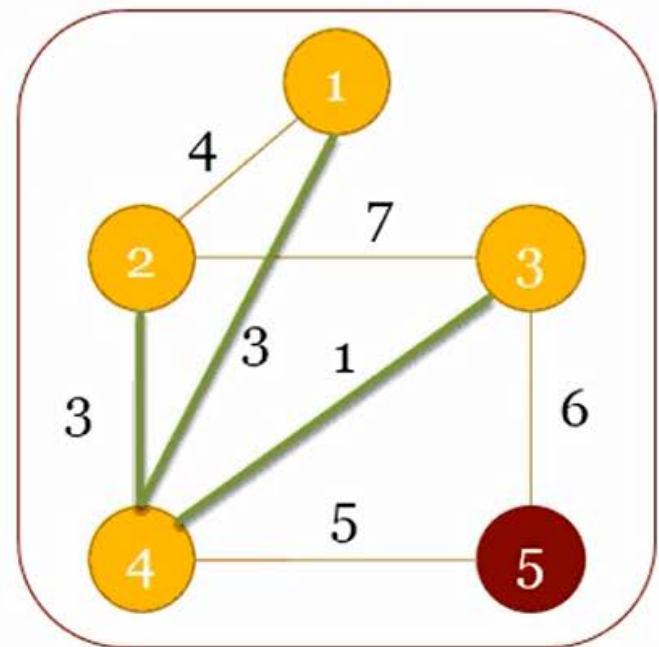
Ví dụ thuật toán Kruskal

- Chọn cạnh $(1, 4)$ – là cạnh tiếp theo trong danh sách.
- $T = \{\{3, 4\}, \{1, 4\}\}$
- Danh sách các cạnh tăng dần:
 - $(3, 4) = 1$
 - $(1, 4) = 3$
 - $(2, 4) = 3$
 - $(1, 2) = 4$
 - $(4, 5) = 5$
 - $(3, 5) = 6$
 - $(2, 3) = 7$



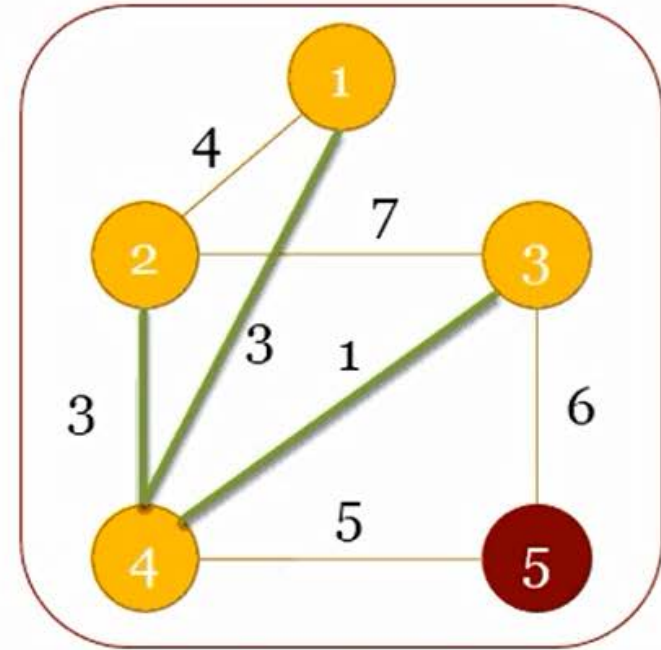
Ví dụ thuật toán Kruskal

- Chọn cạnh (2, 4) – là cạnh tiếp theo trong danh sách.
- $T = \{\{3, 4\}, \{1, 4\}, \{2, 4\}\}$
- Danh sách các cạnh tăng dần:
 - $(3, 4) = 1$
 - $(1, 4) = 3$
 - $(2, 4) = 3$
 - $(1, 2) = 4$
 - $(4, 5) = 5$
 - $(3, 5) = 6$
 - $(2, 3) = 7$



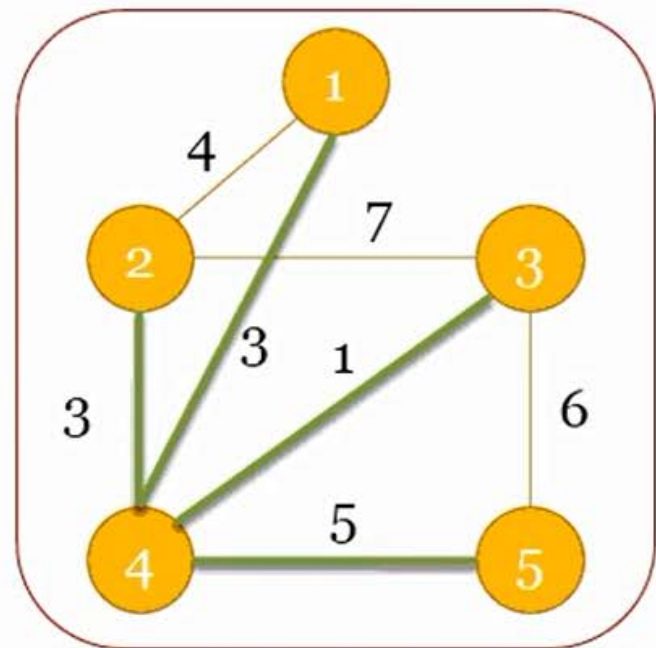
Ví dụ thuật toán Kruskal

- KHÔNG chọn cạnh (1, 2) – là cạnh tiếp theo trong danh sách vì khi chọn sẽ tạo thành chu trình.
- $T = \{(3, 4), (1, 4), (2, 4)\}$
- Danh sách các cạnh tăng dần:
 - $(3, 4) = 1$
 - $(1, 4) = 3$
 - $(2, 4) = 3$
 - **$(1, 2) = 4$ không chọn!**
 - $(4, 5) = 5$
 - $(3, 5) = 6$
 - $(2, 3) = 7$



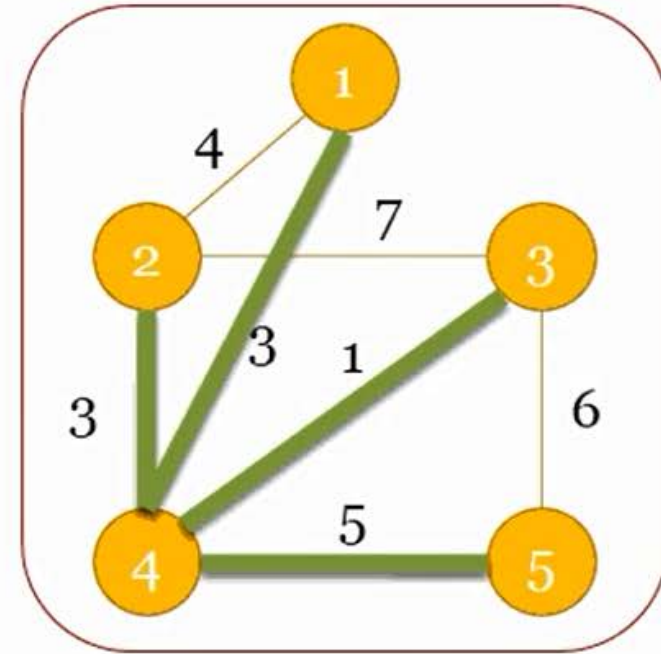
Ví dụ thuật toán Kruskal

- Chọn cạnh (4, 5) – là cạnh tiếp theo trong danh sách.
- $T = \{(3, 4), (1, 4), (2, 4), (4, 5)\}$
- Danh sách các cạnh tăng dần:
 - $(3, 4) = 1$
 - $(1, 4) = 3$
 - $(2, 4) = 3$
 - $(1, 2) = 4$
 - $(4, 5) = 5$
 - $(3, 5) = 6$
 - $(2, 3) = 7$

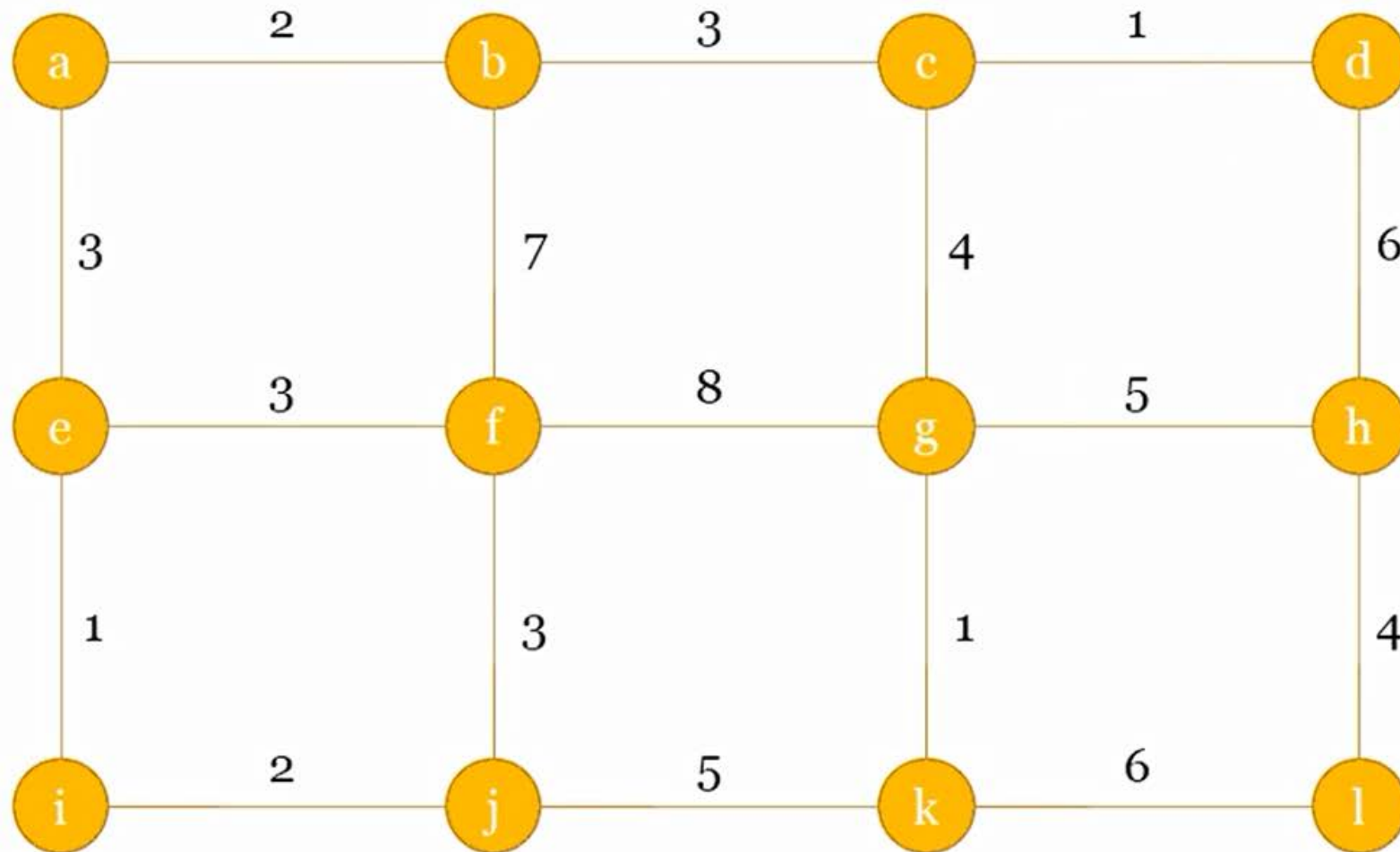


Ví dụ thuật toán Kruskal

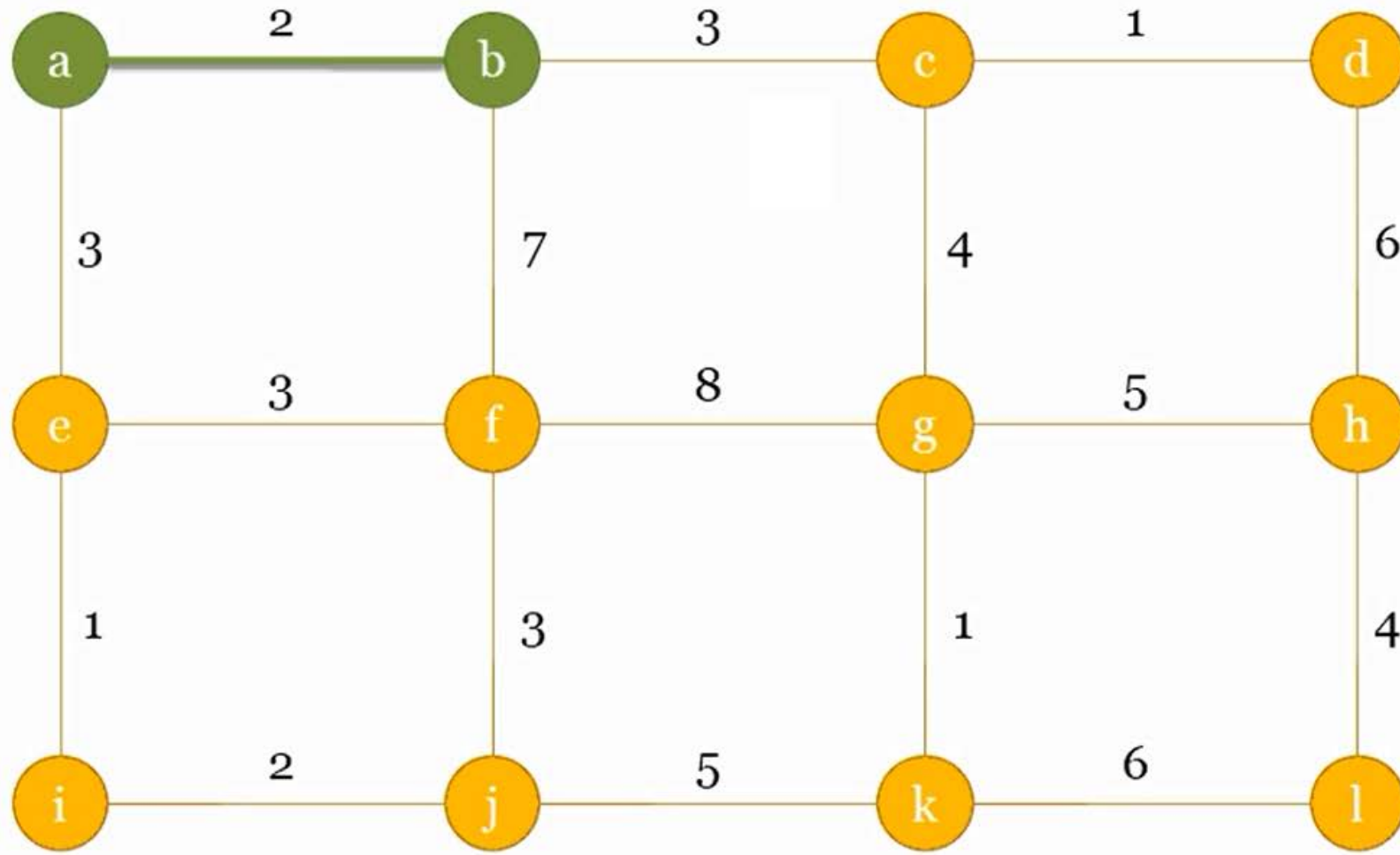
- Do $|T| = 4$, kết thúc thuật toán.
- Cây khung tìm được là:
 - $T = \{\{1, 4\}, \{2, 4\}, \{3, 4\}, \{4, 5\}\}$



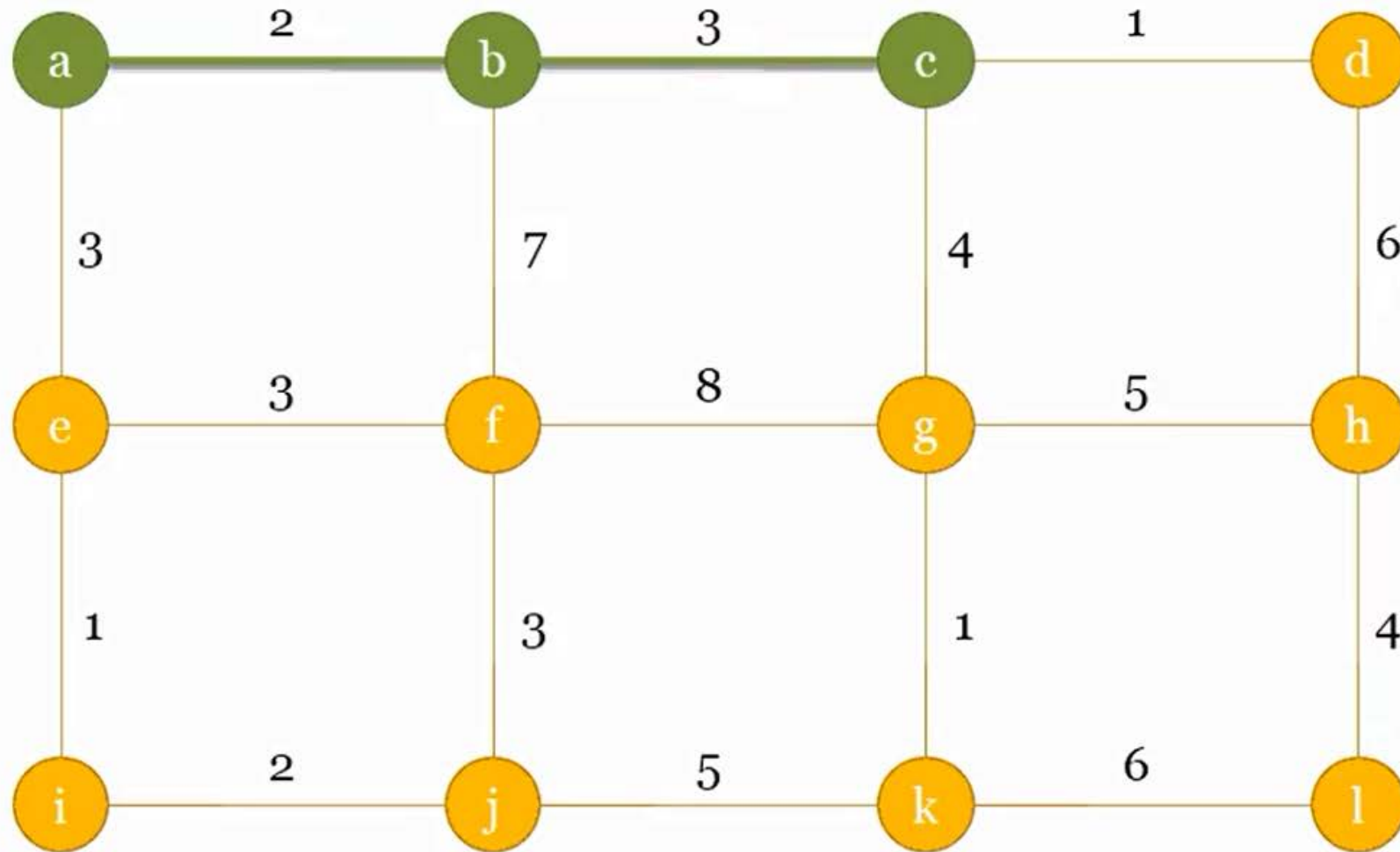
Ví dụ tìm cây khung nhỏ nhất



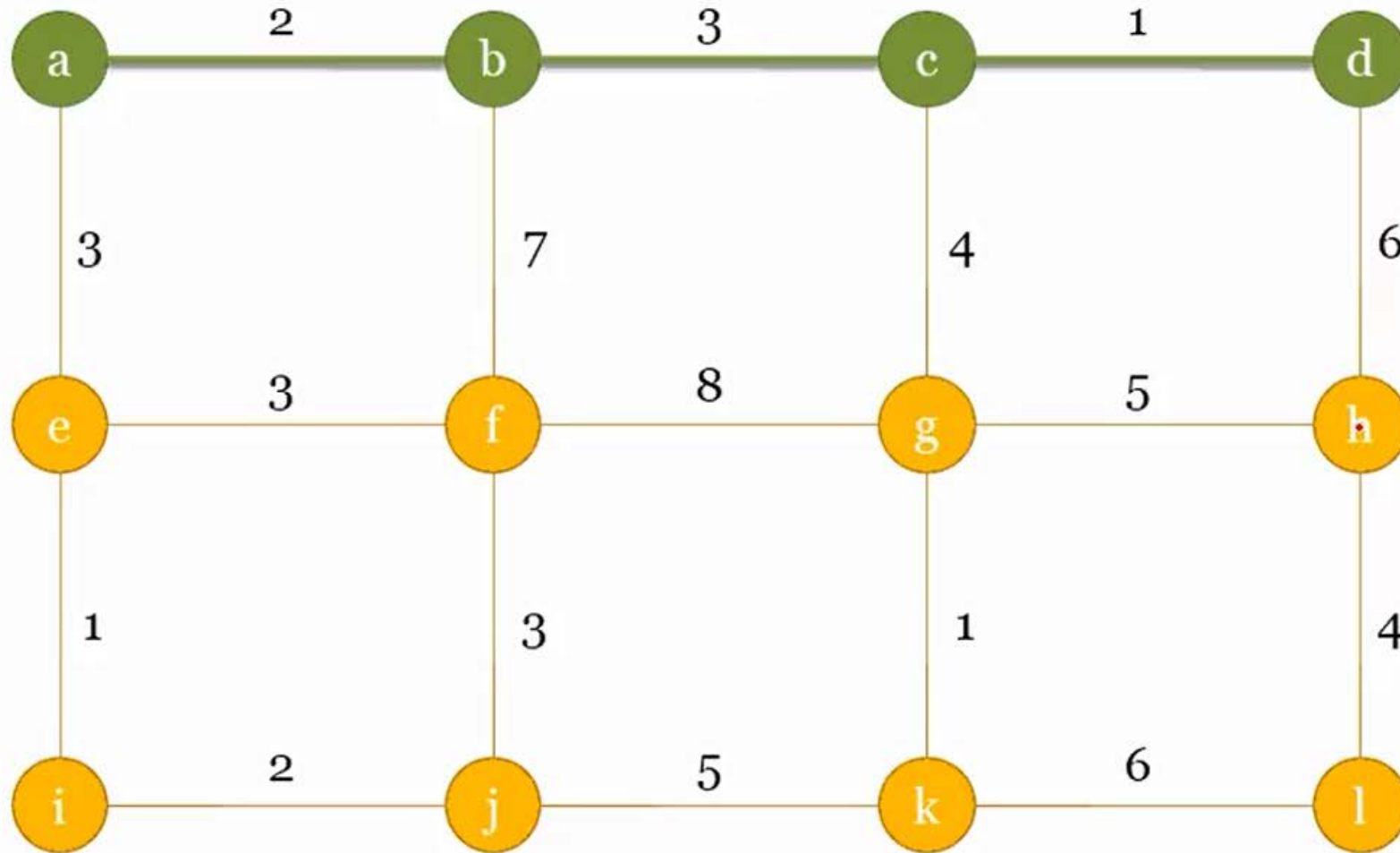
Tìm cây khung nhỏ nhất theo Prim



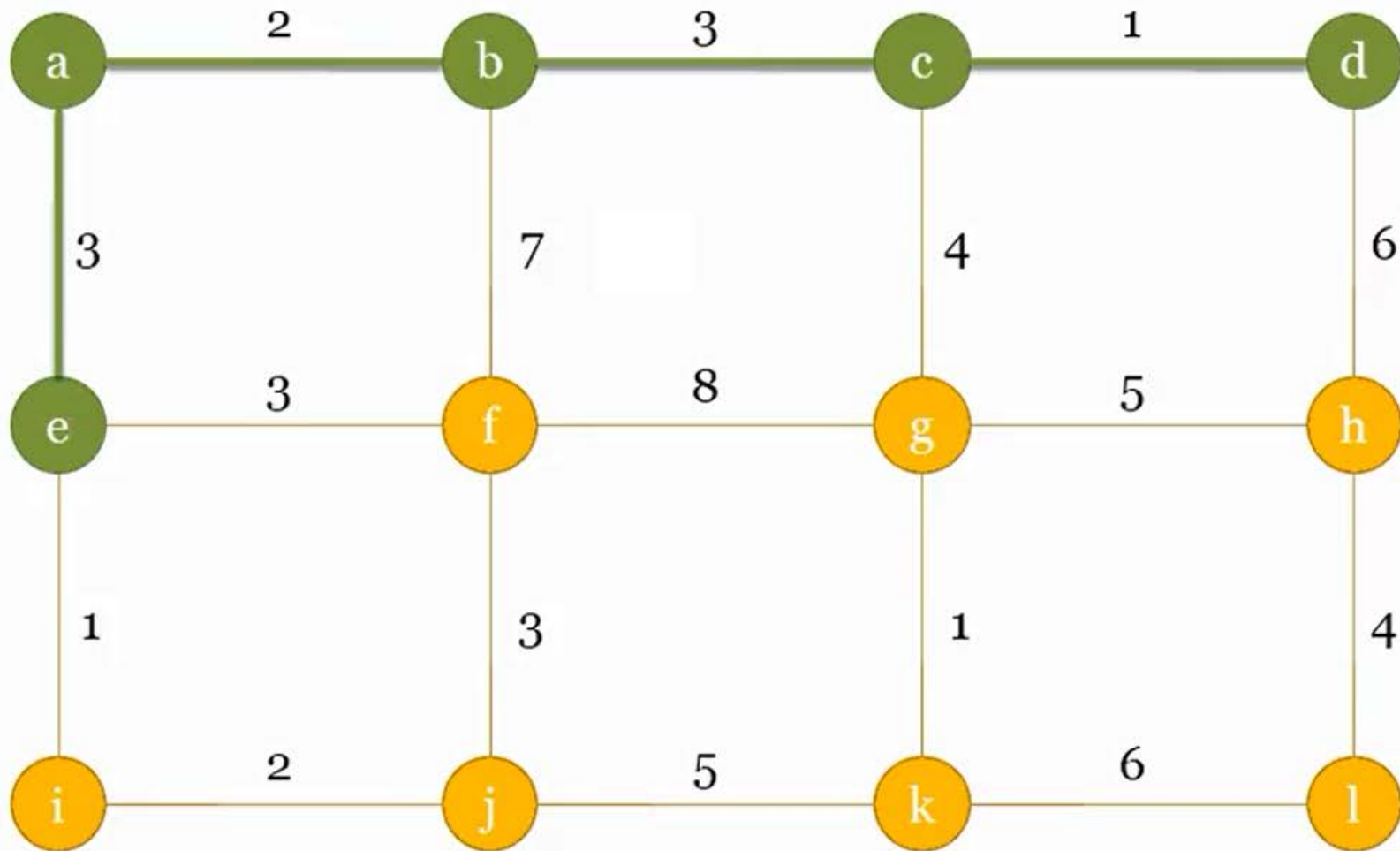
Tìm cây khung nhỏ nhất theo Prim



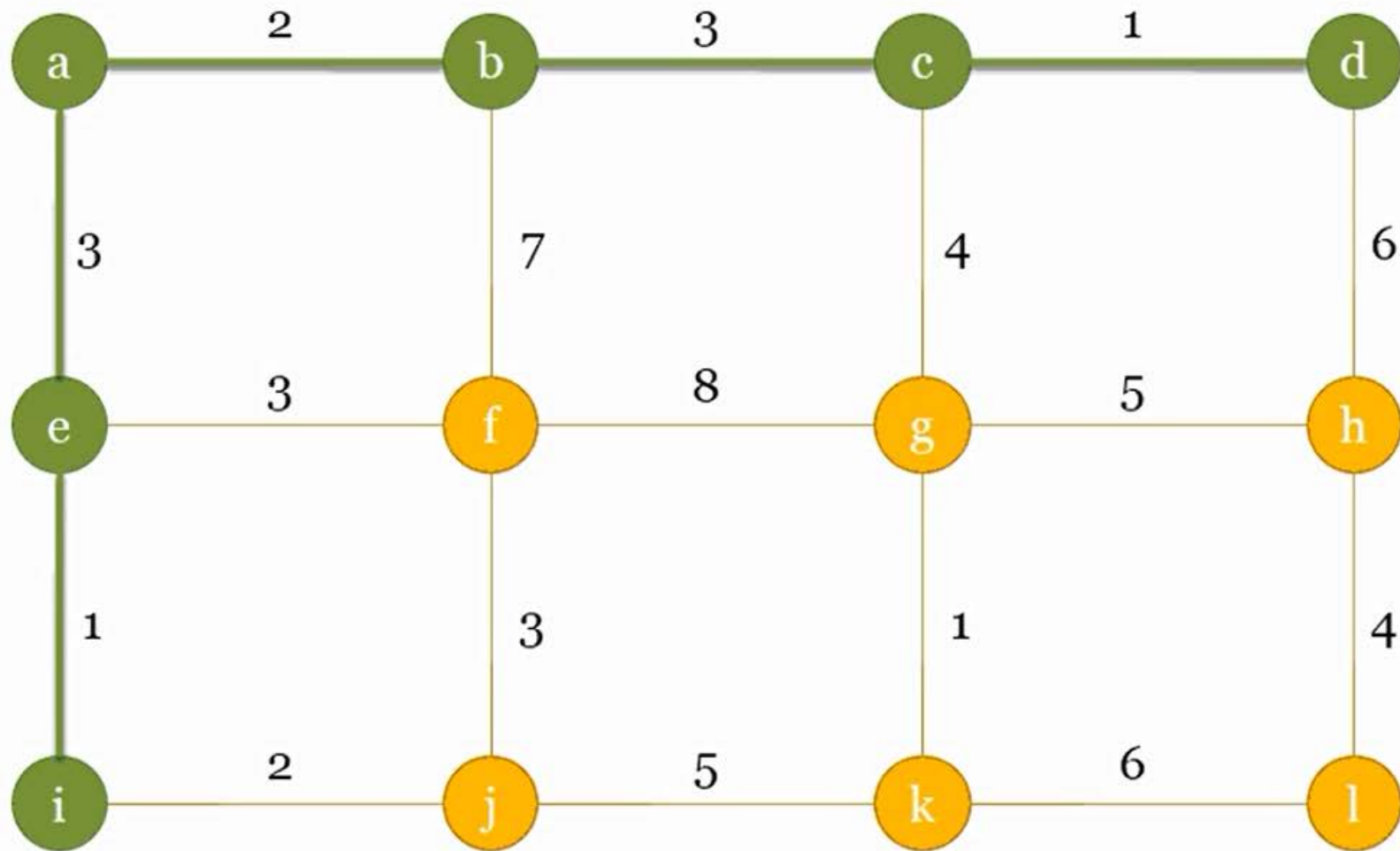
Tìm cây khung nhỏ nhất theo Prim



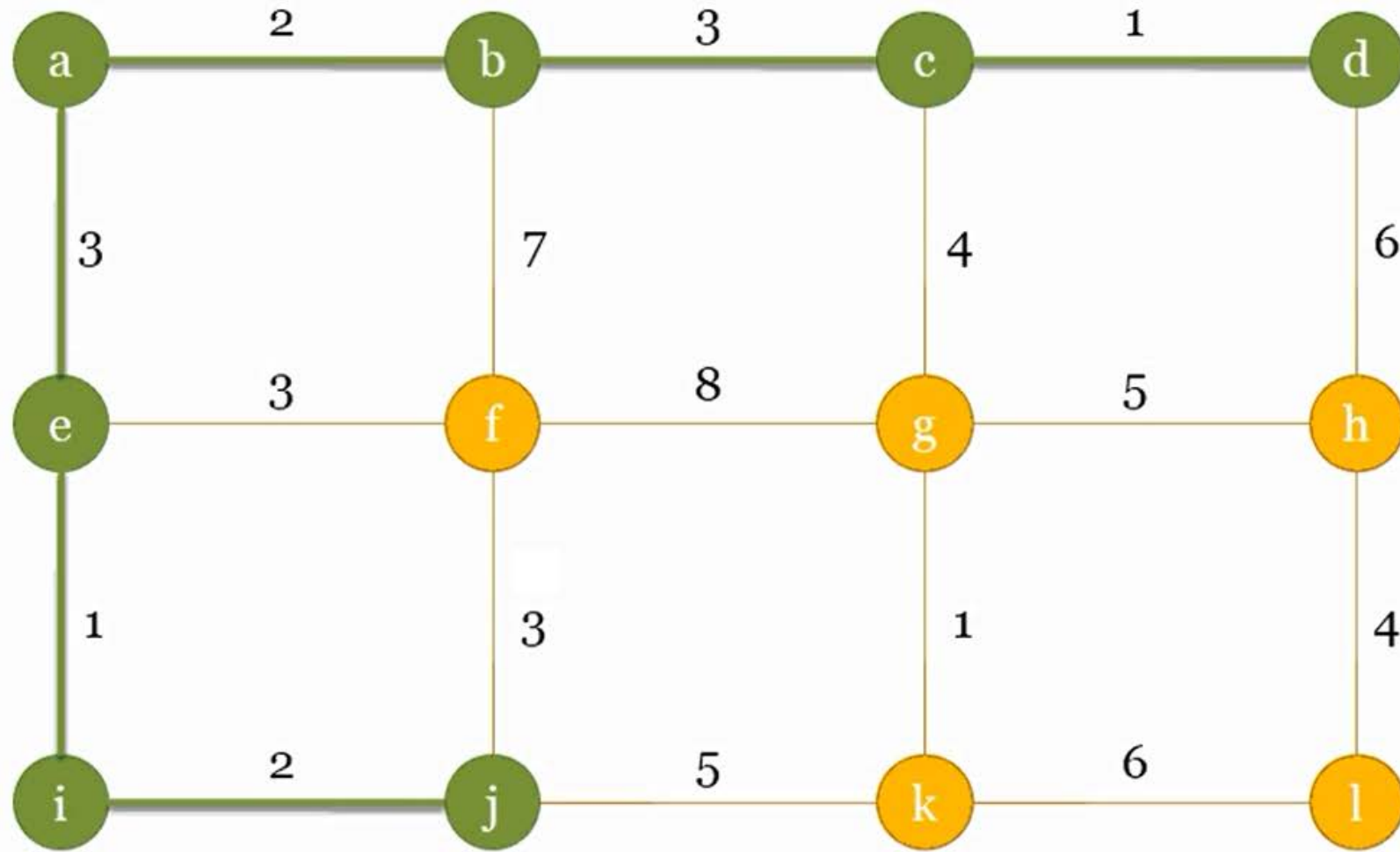
Tìm cây khung nhỏ nhất theo Prim



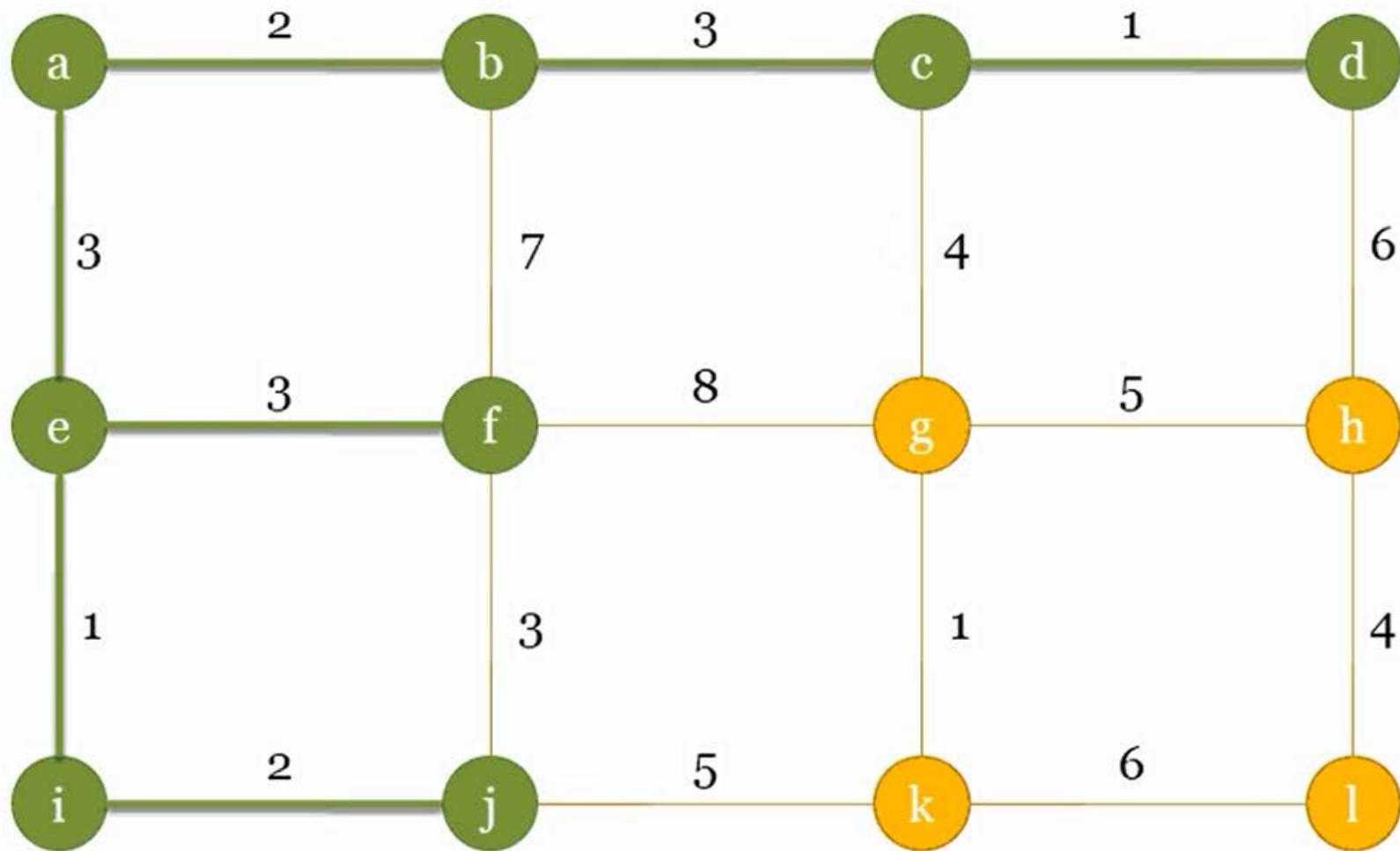
Tìm cây khung nhỏ nhất theo Prim



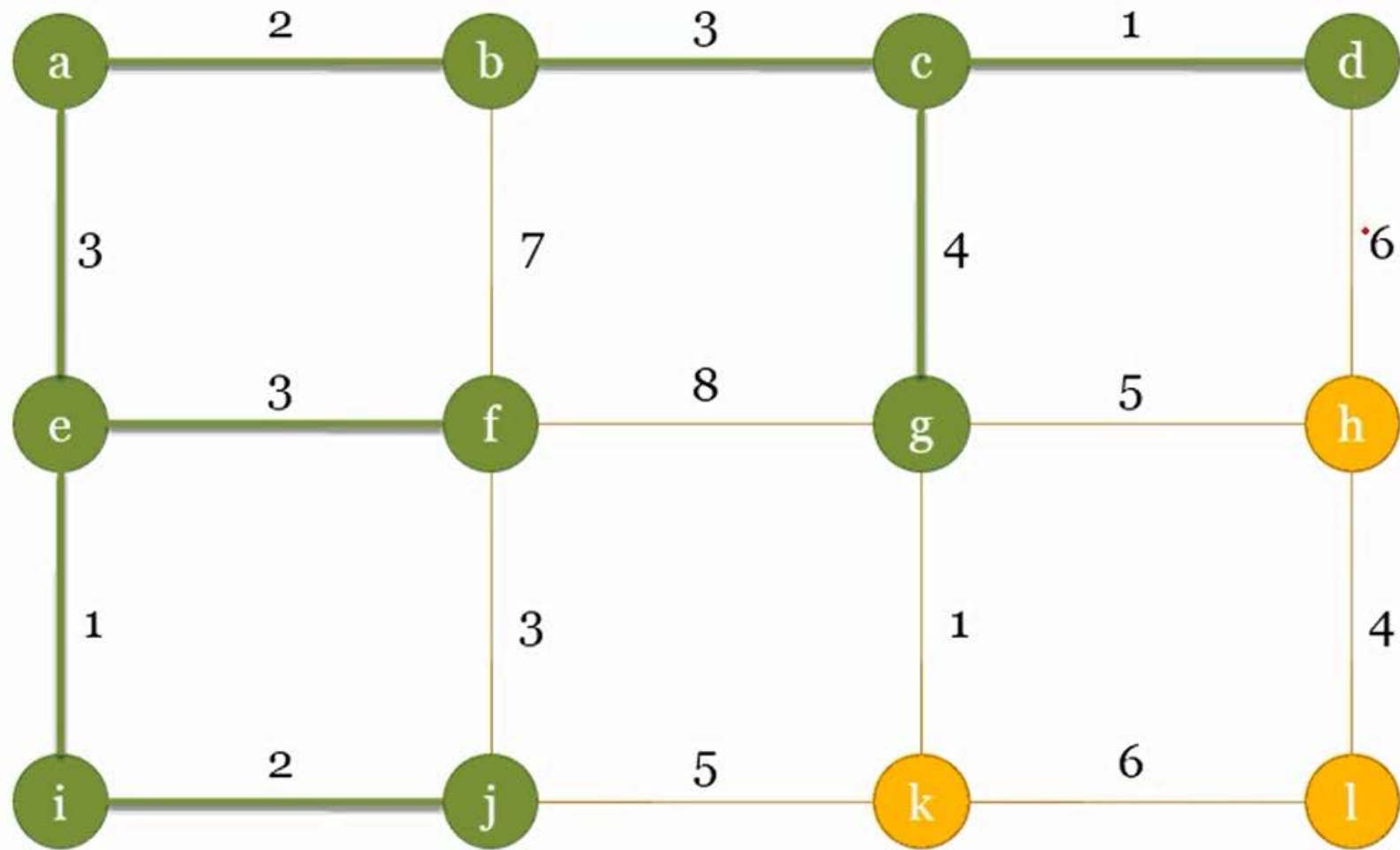
Tìm cây khung nhỏ nhất theo Prim



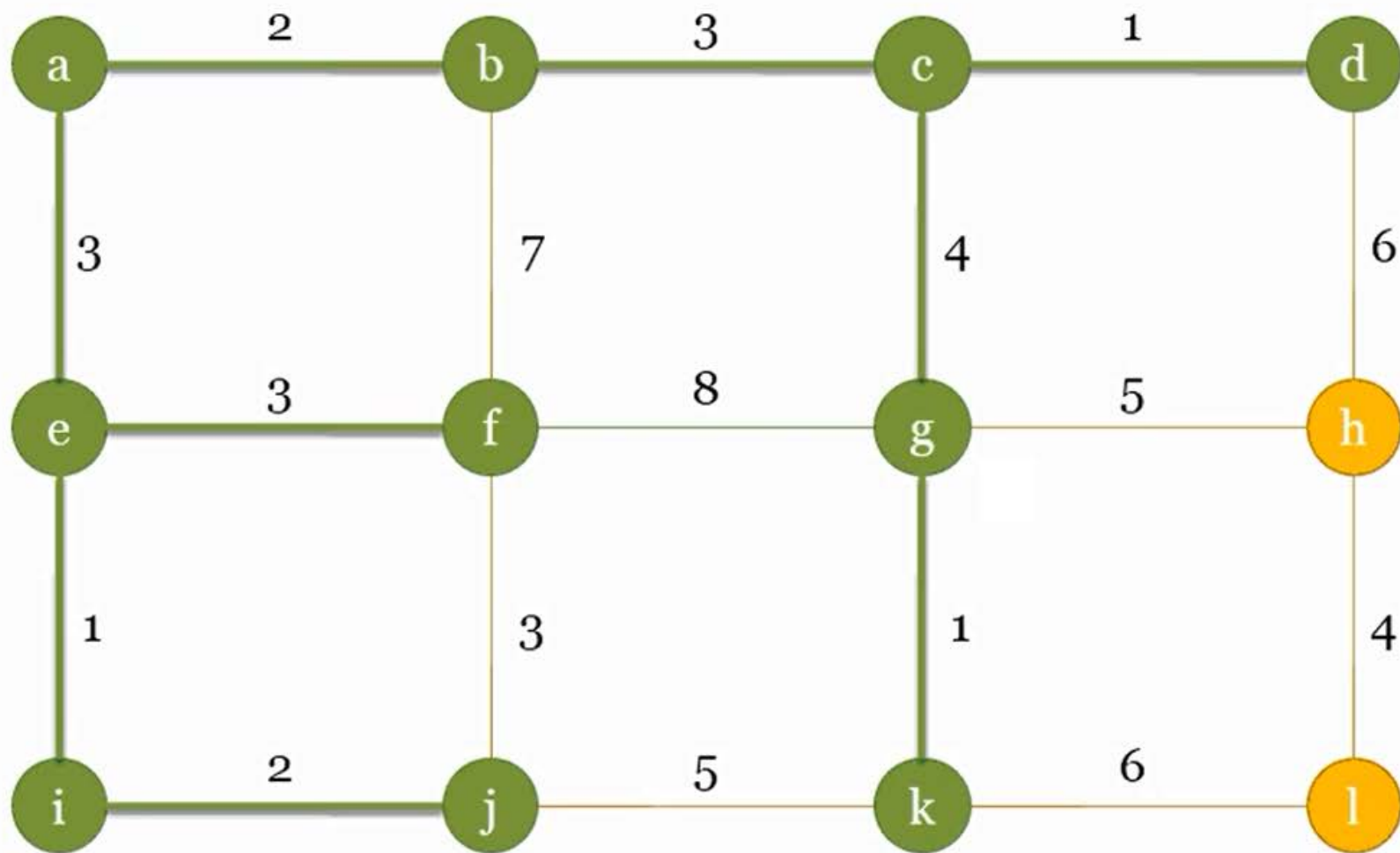
Tìm cây khung nhỏ nhất theo Prim



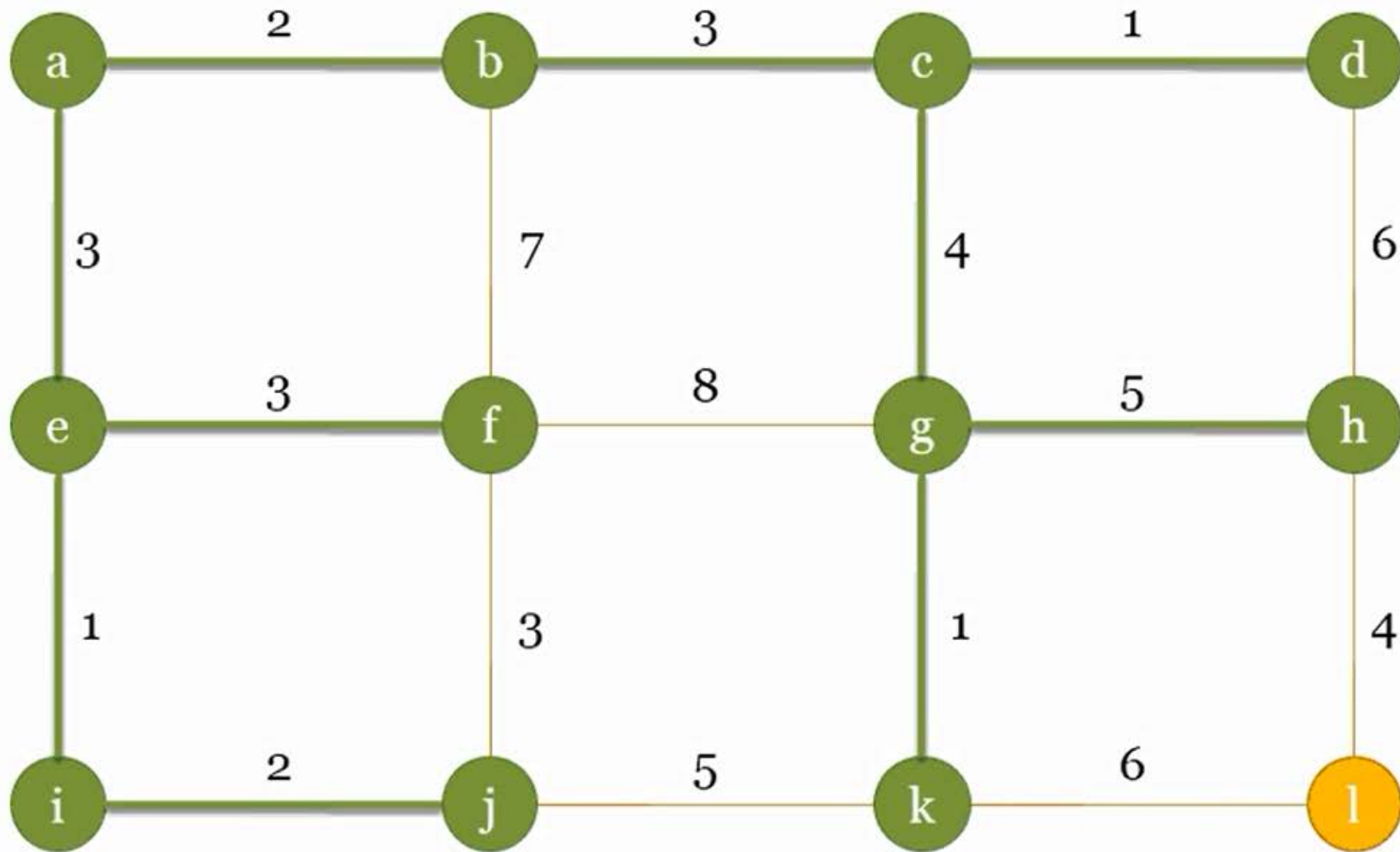
Tìm cây khung nhỏ nhất theo Prim



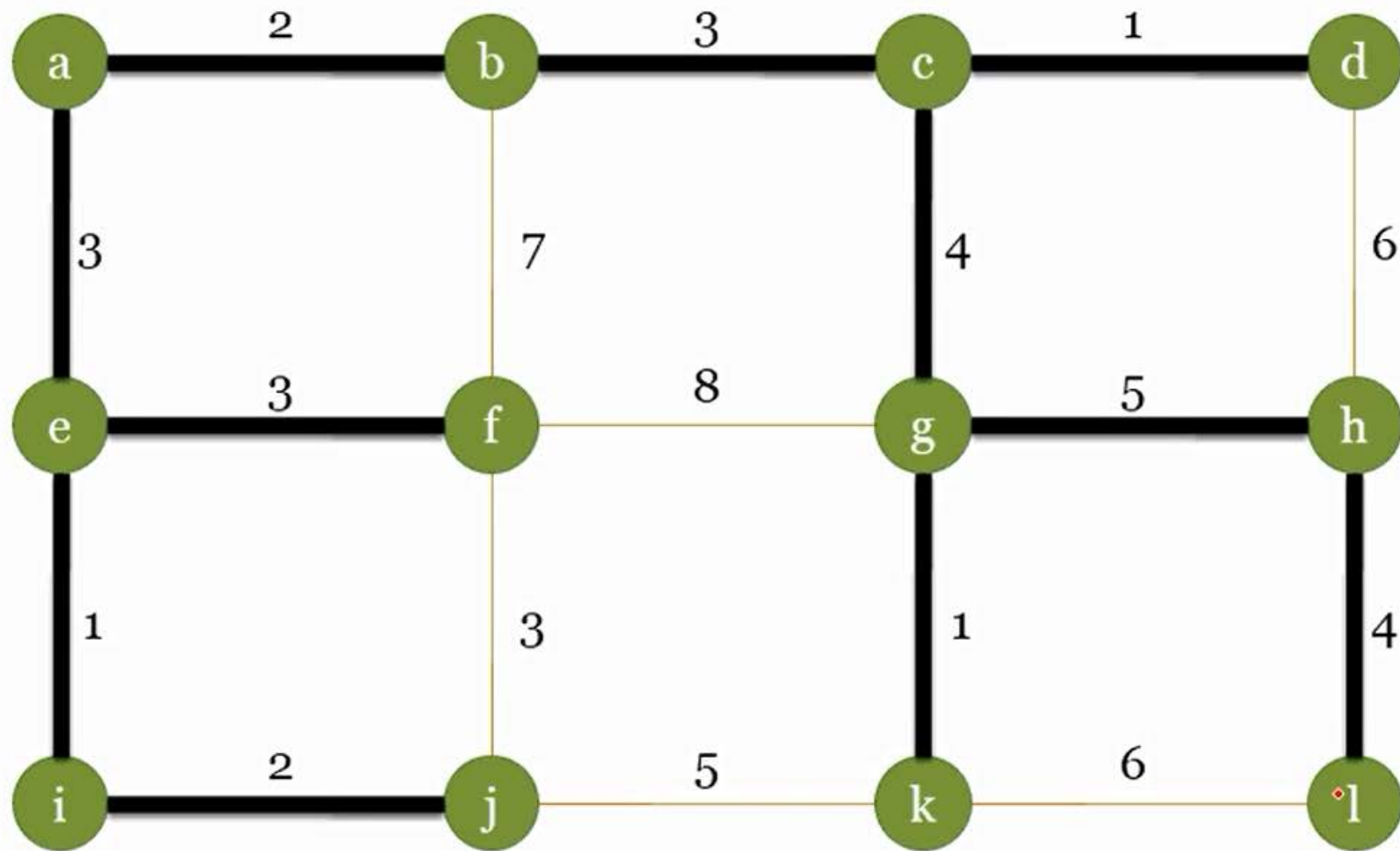
Tìm cây khung nhỏ nhất theo Prim



Tìm cây khung nhỏ nhất theo Prim

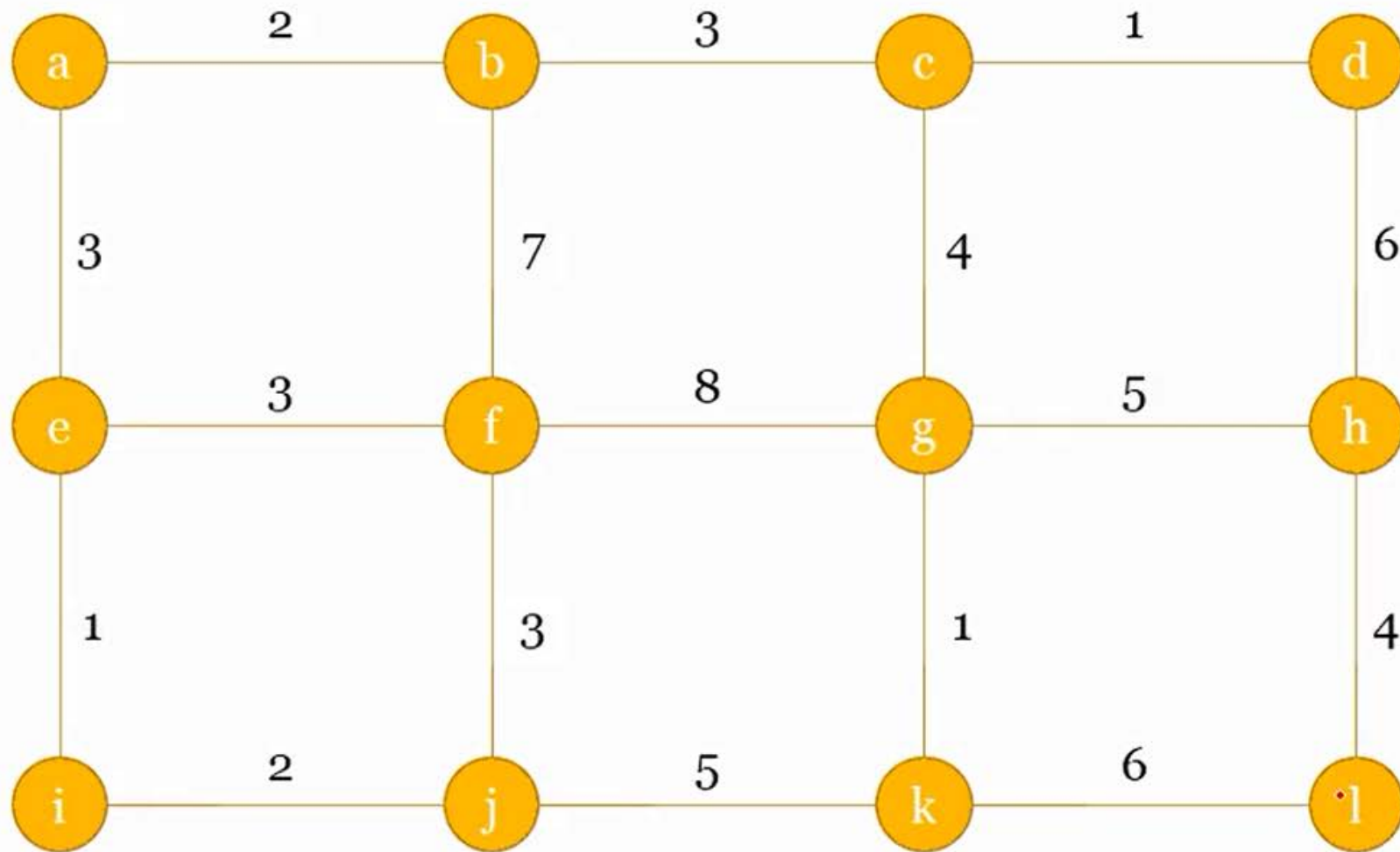


Tìm cây khung nhỏ nhất theo Prim

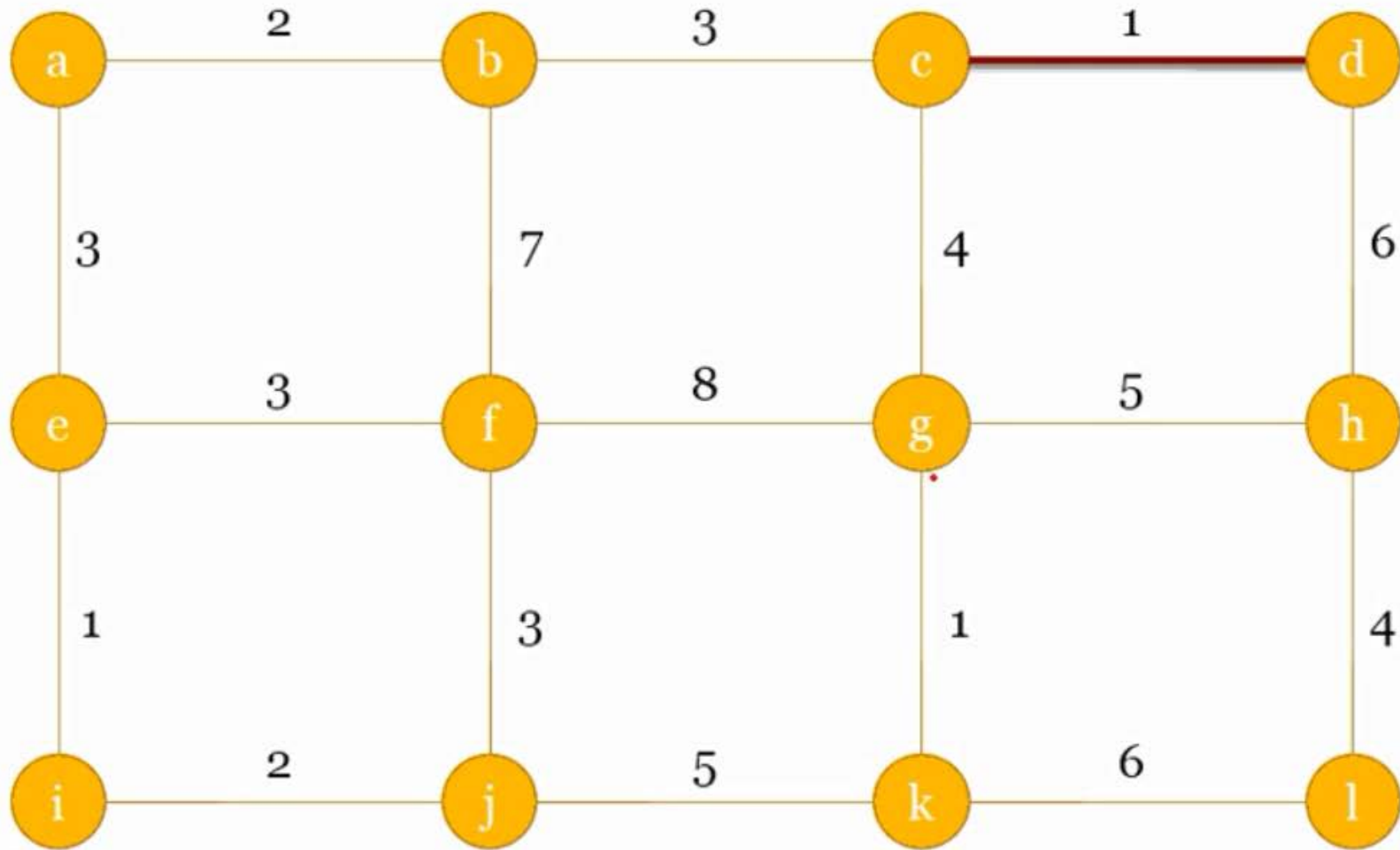


$$c(T) = 29$$

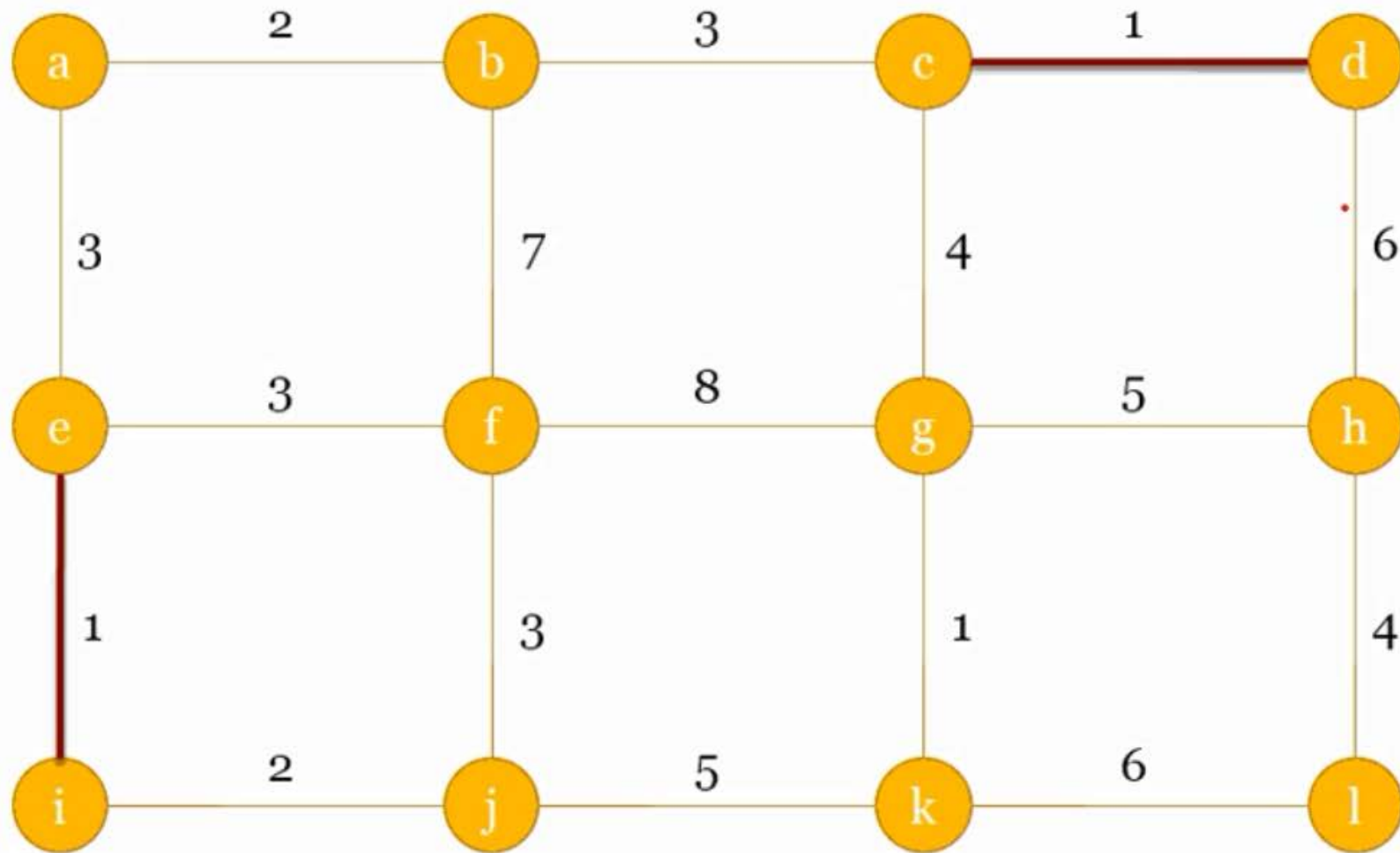
Tìm cây khung nhỏ nhất theo Kruskal



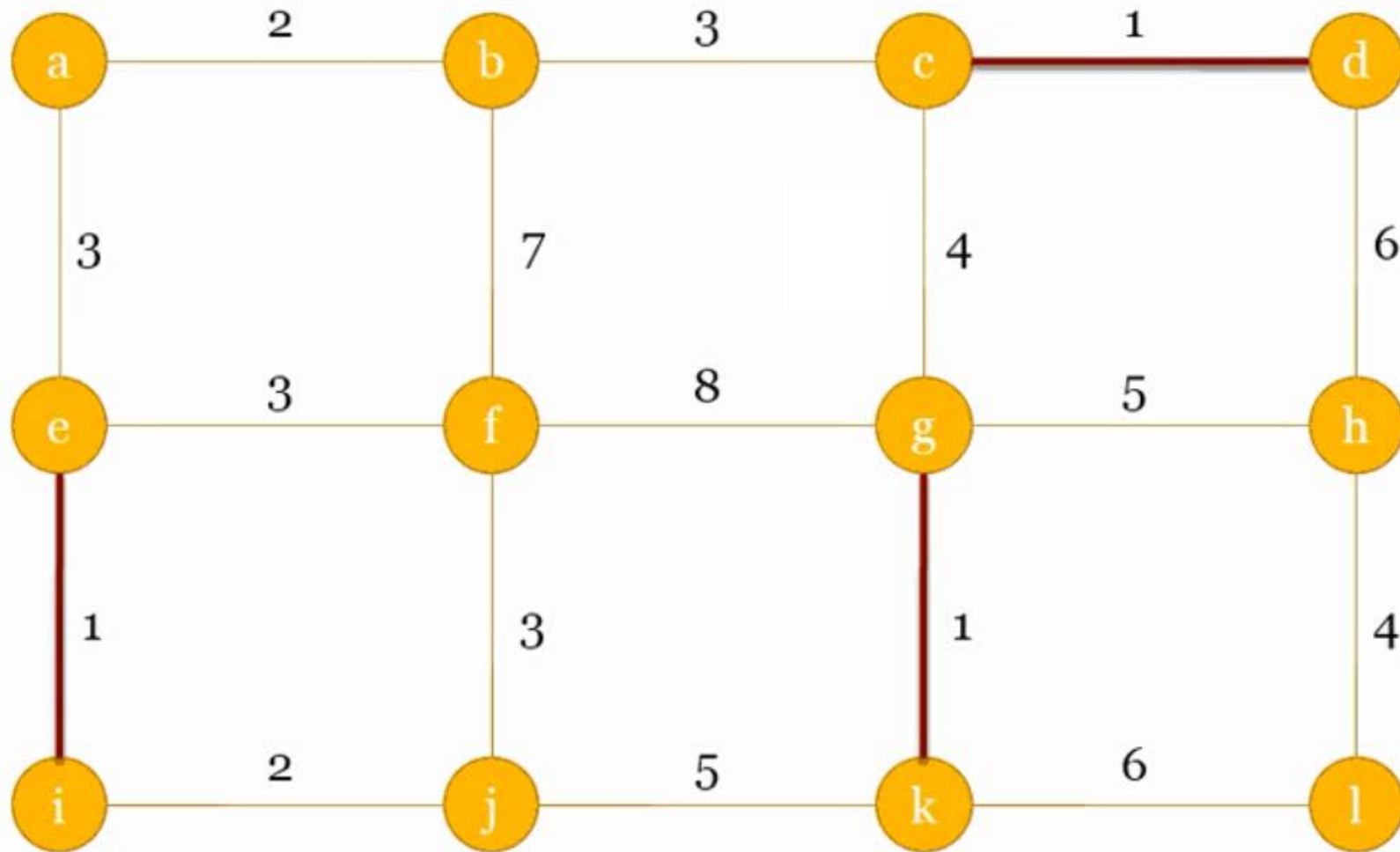
Tìm cây khung nhỏ nhất theo Kruskal



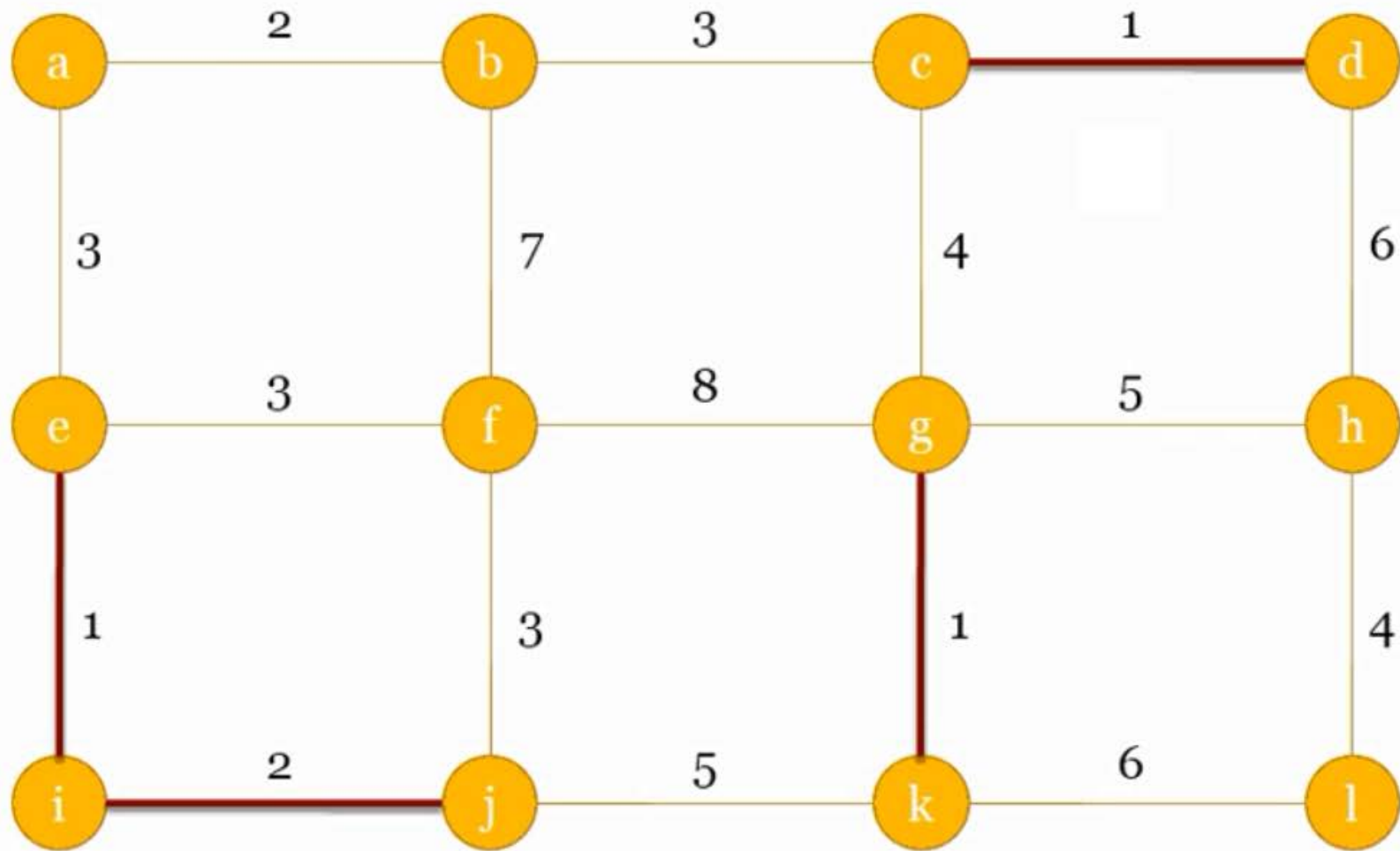
Tìm cây khung nhỏ nhất theo Kruskal



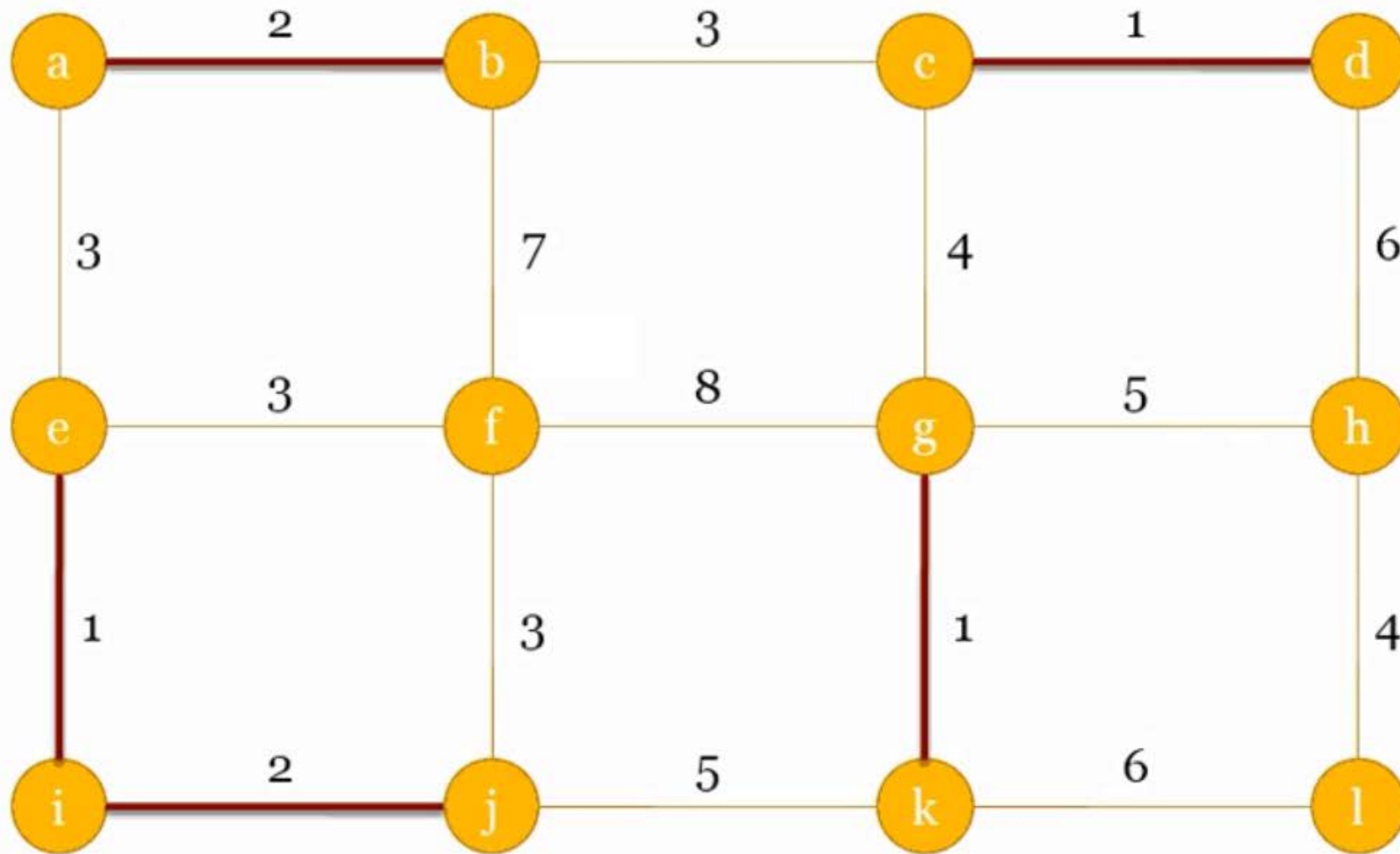
Tìm cây khung nhỏ nhất theo Kruskal



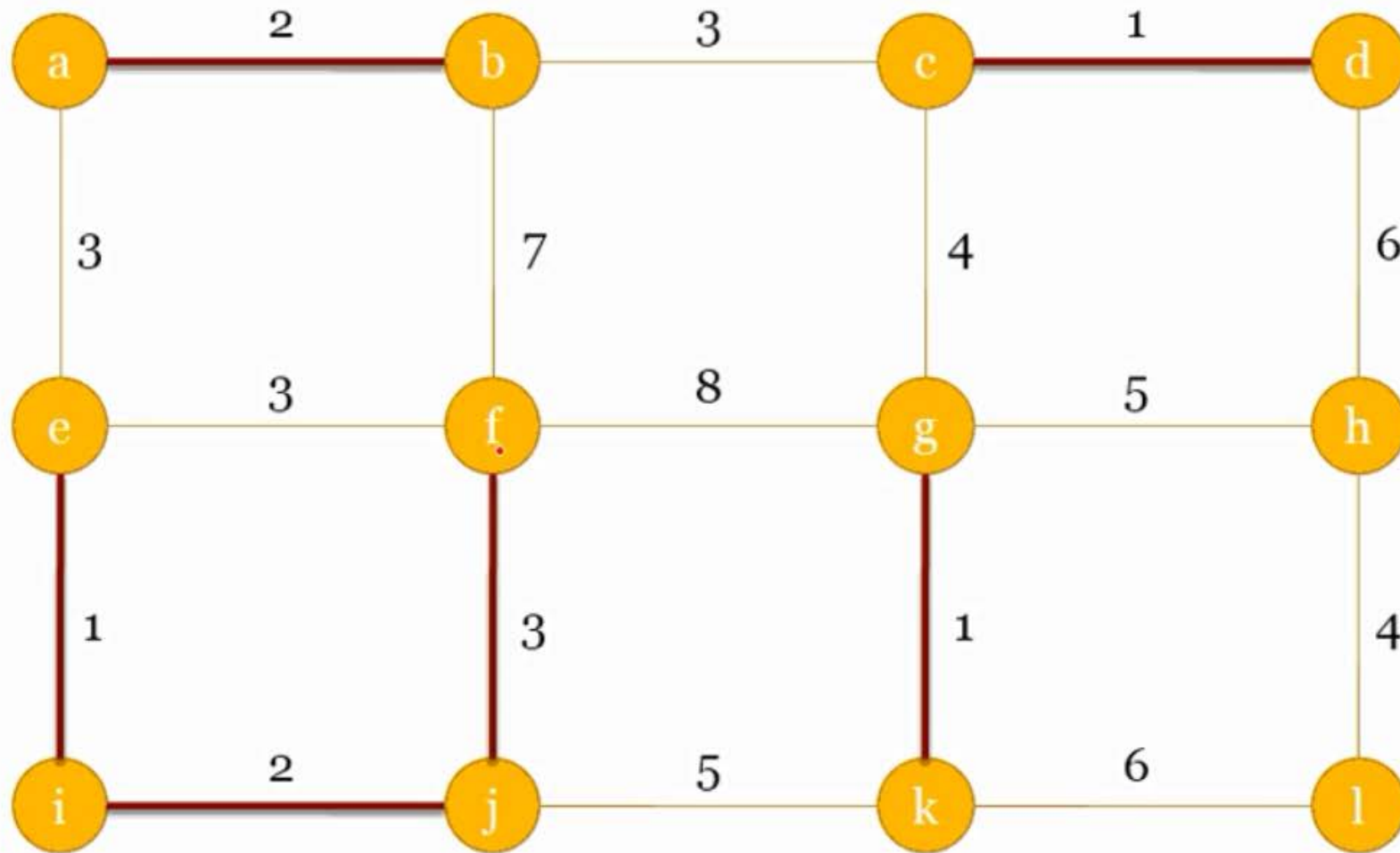
Tìm cây khung nhỏ nhất theo Kruskal



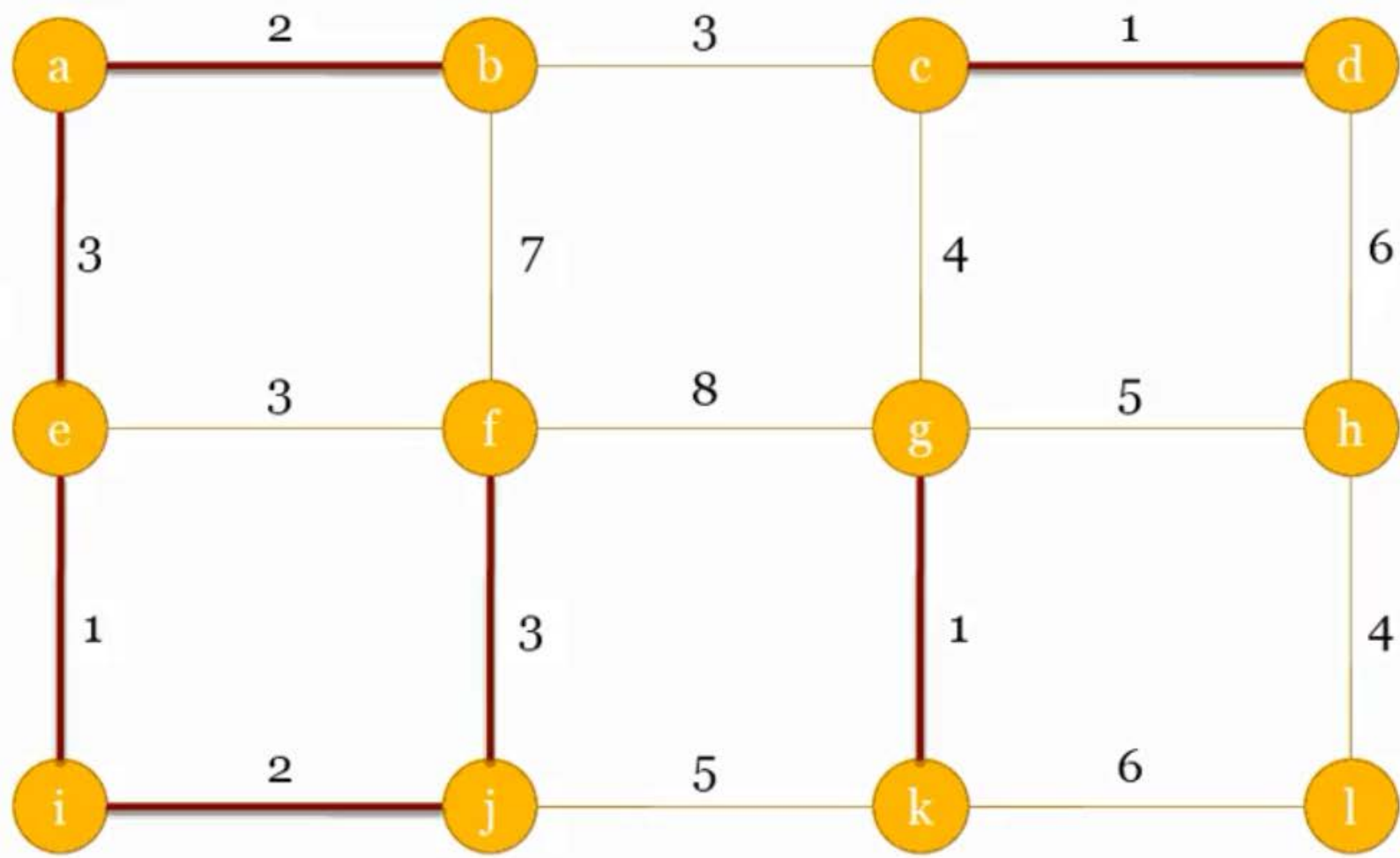
Tìm cây khung nhỏ nhất theo Kruskal



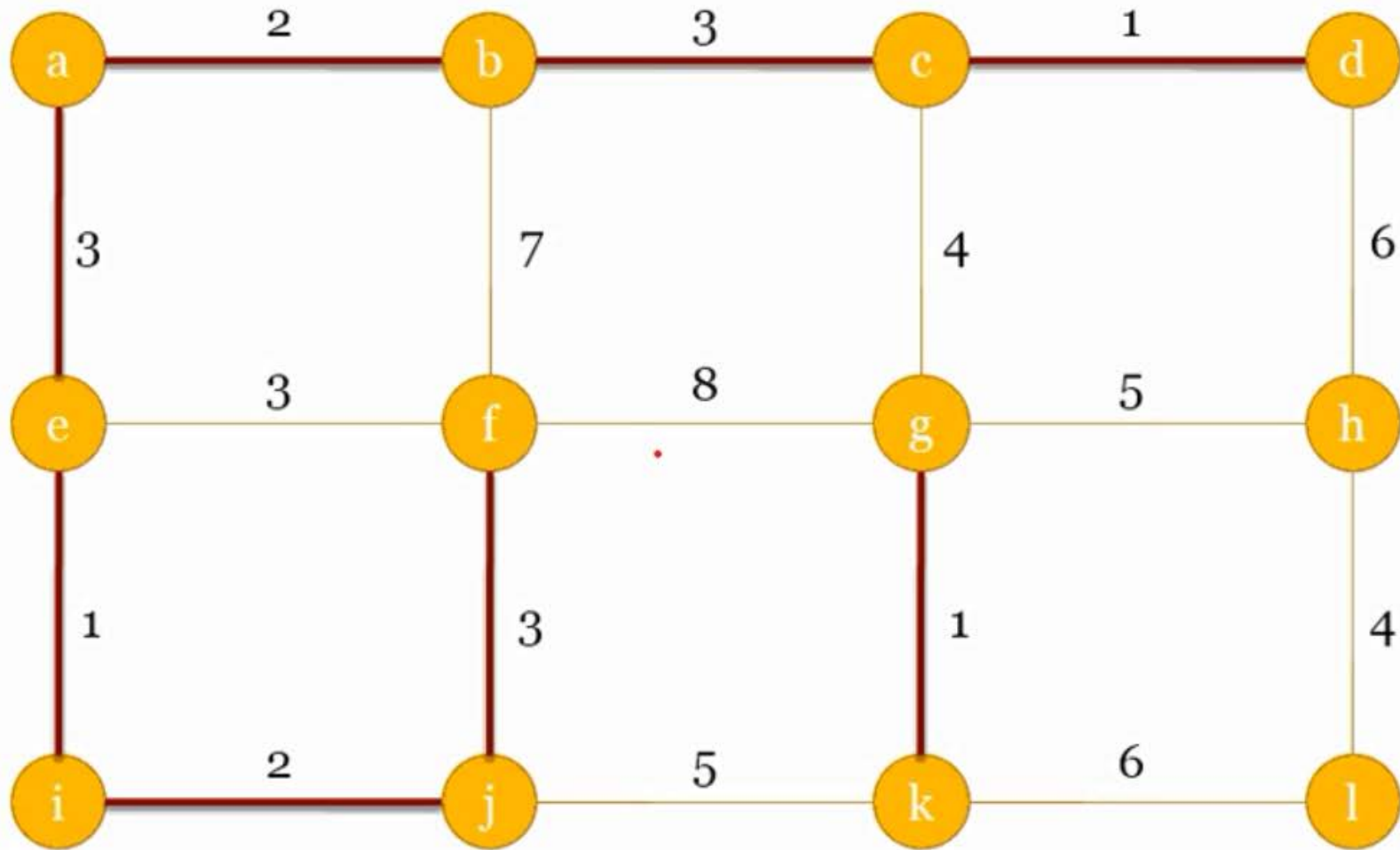
Tìm cây khung nhỏ nhất theo Kruskal



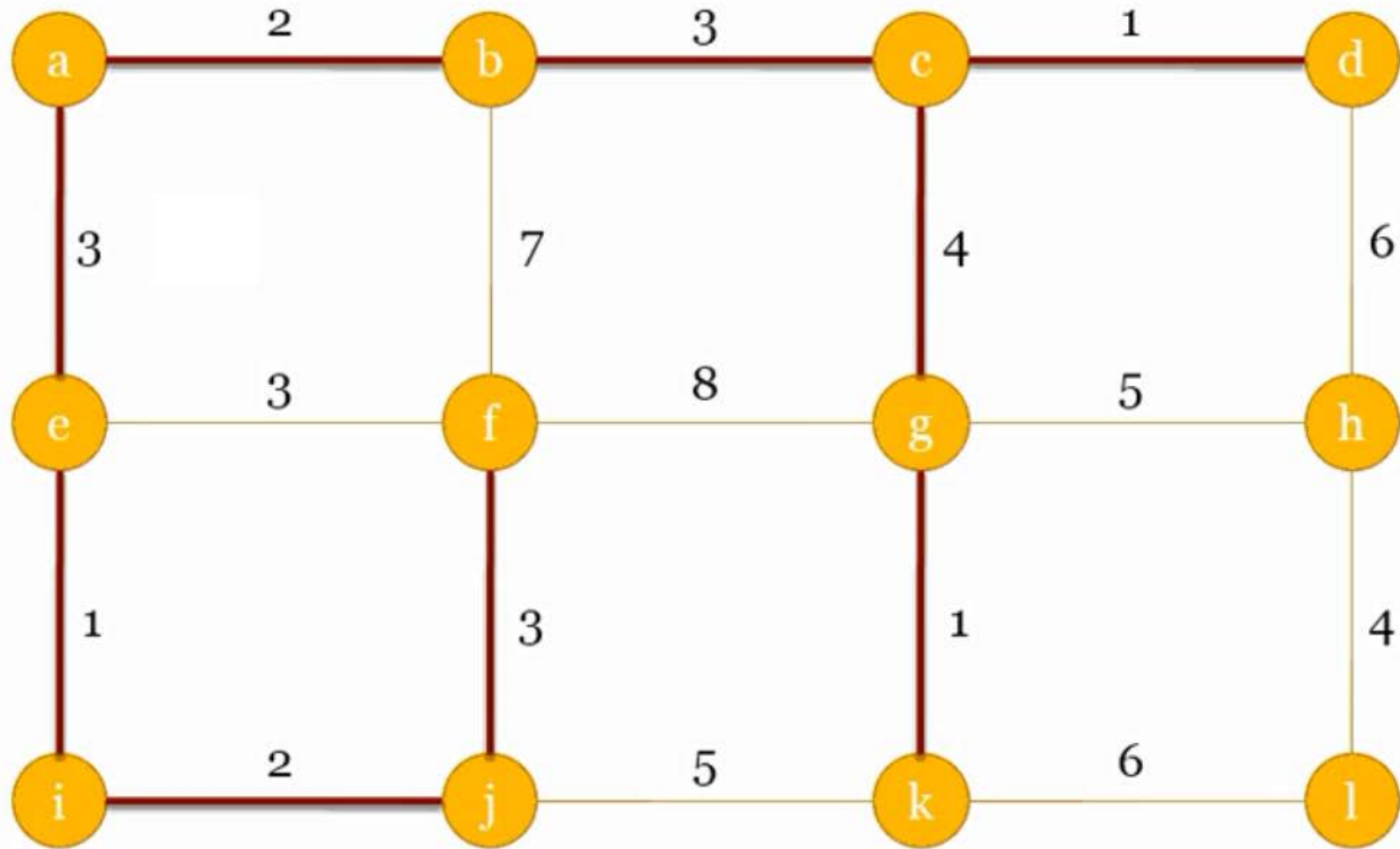
Tìm cây khung nhỏ nhất theo Kruskal



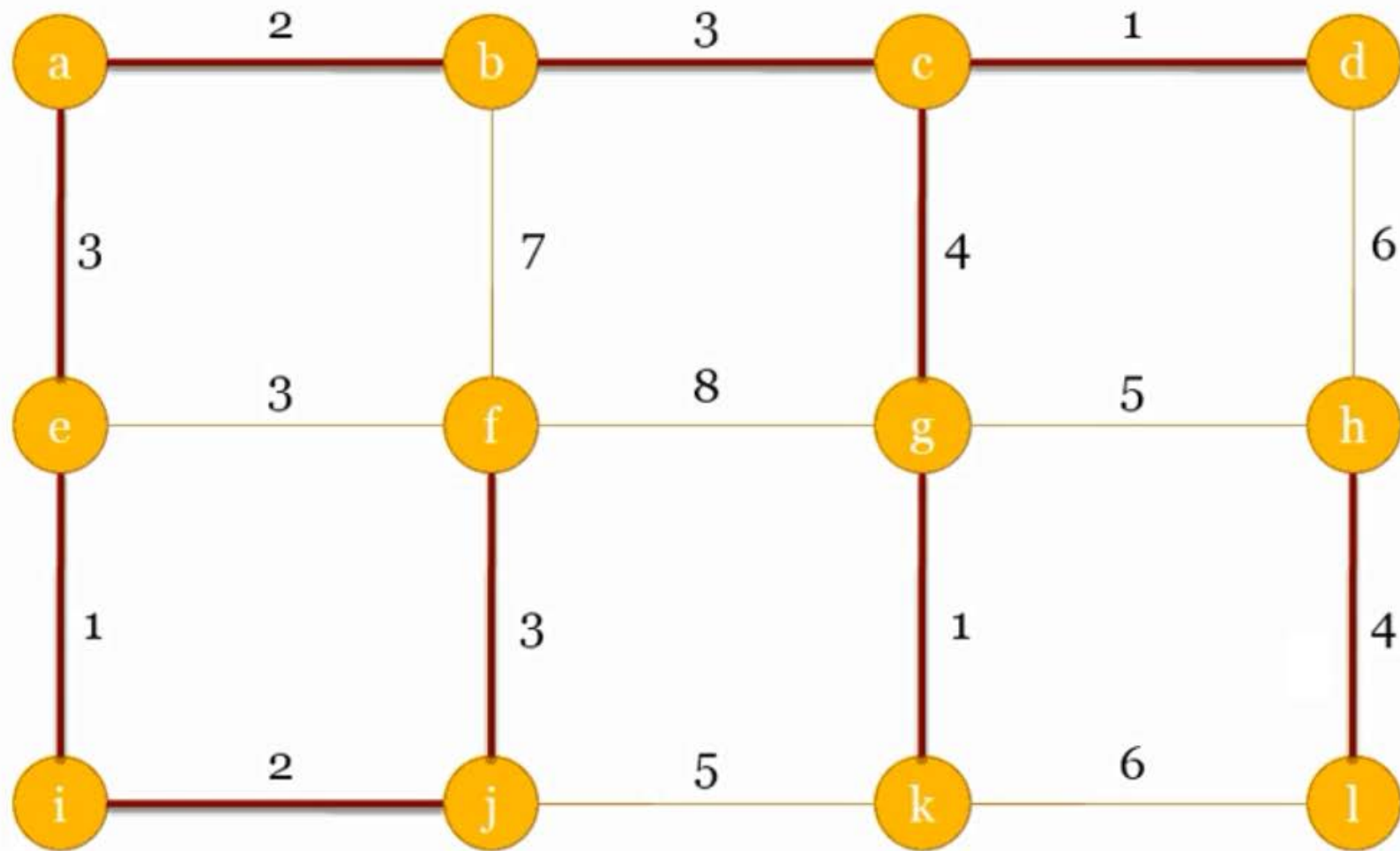
Tìm cây khung nhỏ nhất theo Kruskal



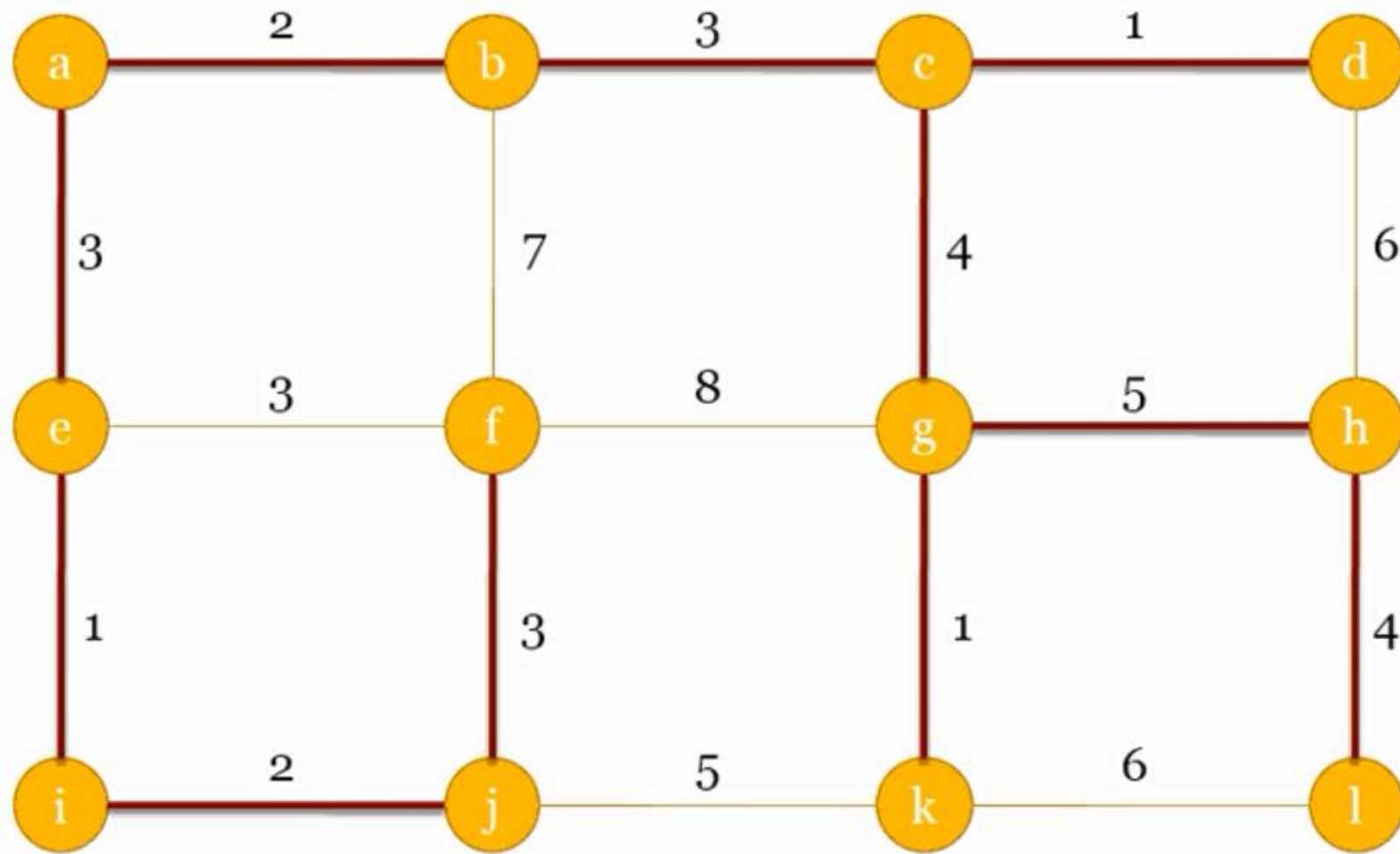
Tìm cây khung nhỏ nhất theo Kruskal



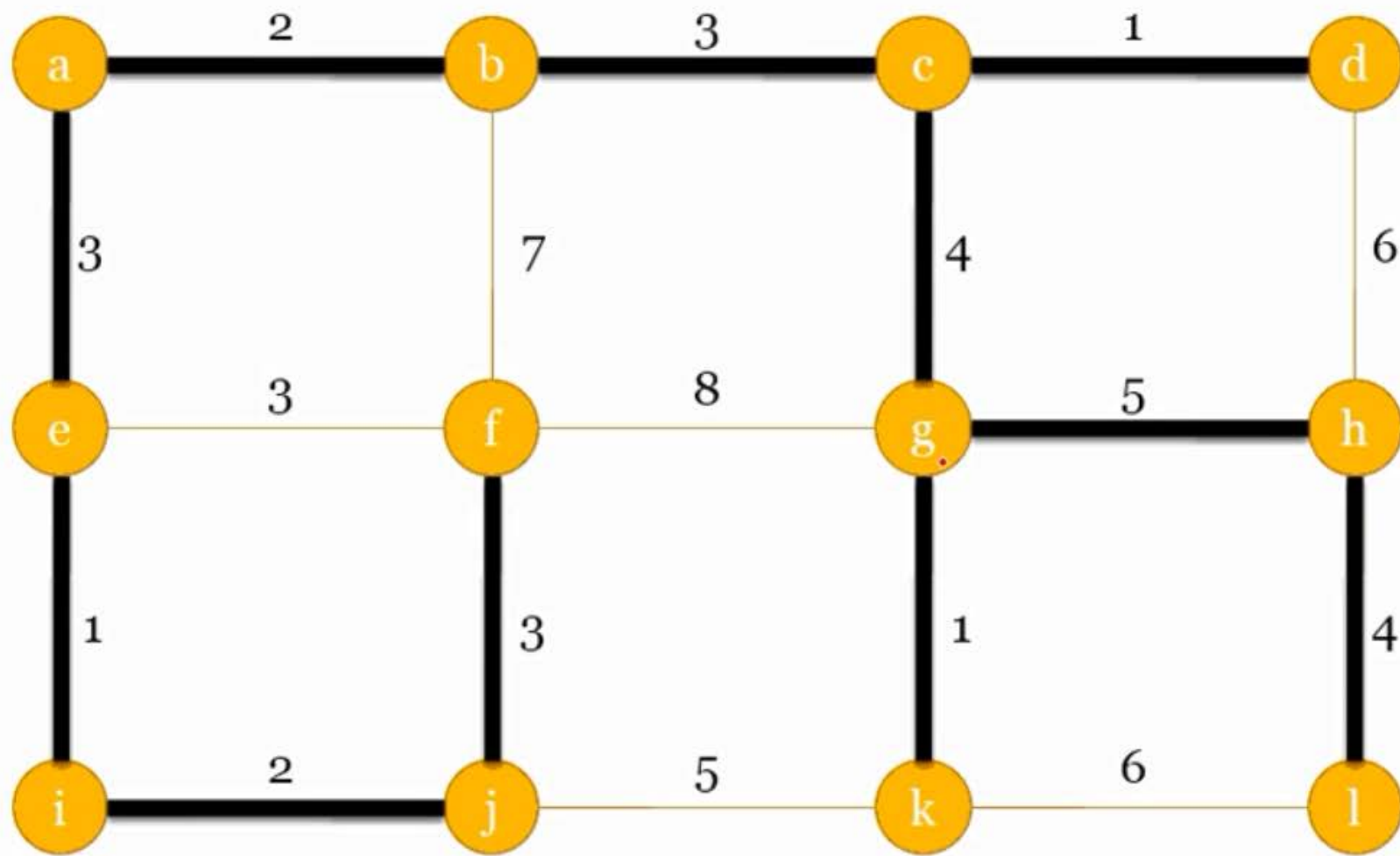
Tìm cây khung nhỏ nhất theo Kruskal



Tìm cây khung nhỏ nhất theo Kruskal

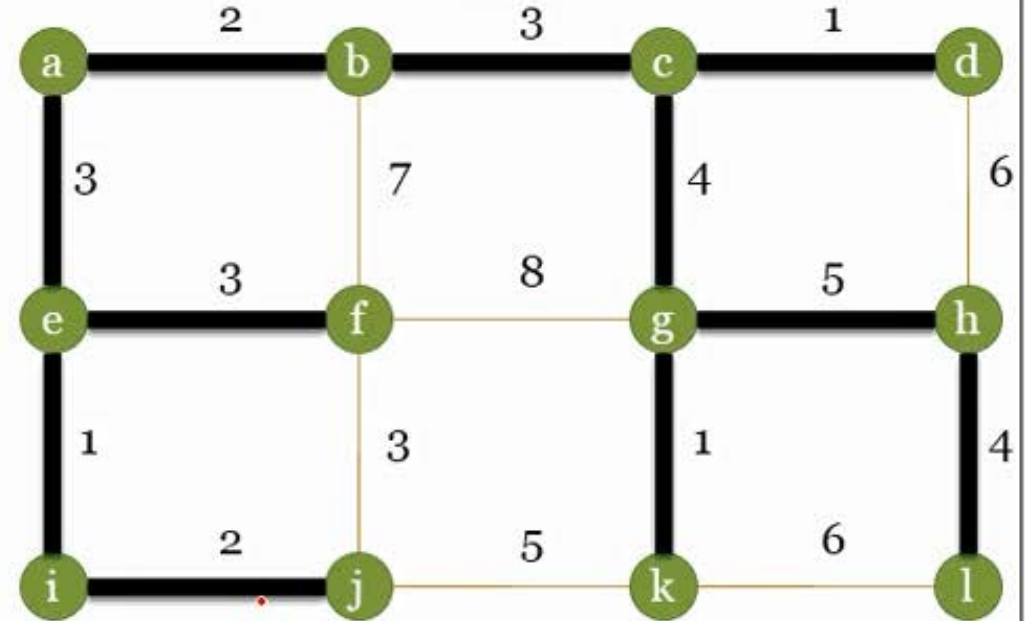
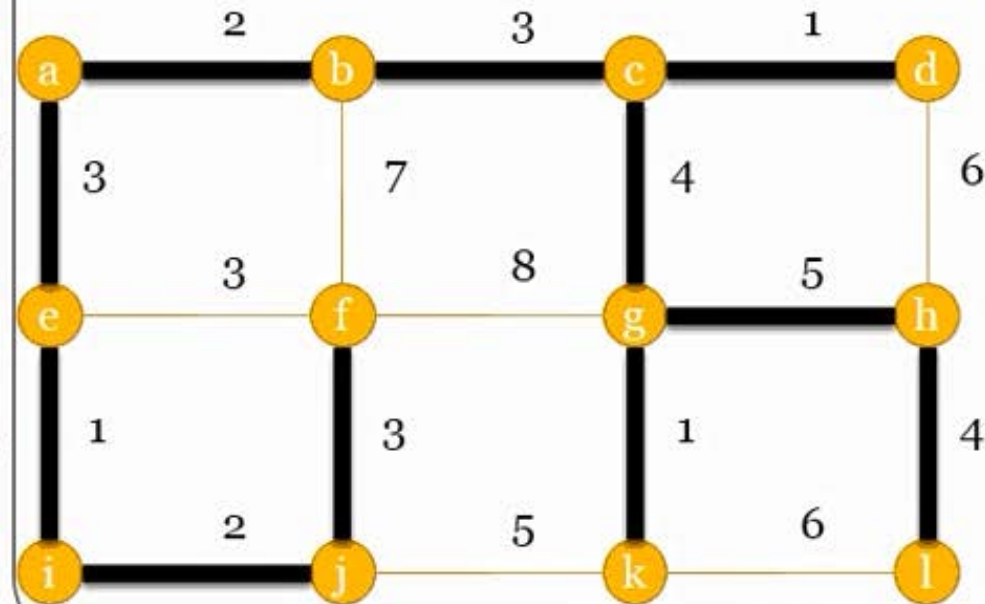


Tìm cây khung nhỏ nhất theo Kruskal



$$c(T) = 29$$

Kết quả Tìm cây khung nhỏ nhất



Tìm cây khung lớn nhất

- Ta có thể tìm cây khung lớn nhất bằng cách áp dụng một trong hai cách làm sau:
 - Đổi dấu trọng số các cạnh của đồ thị từ dương thành âm sau đó tìm cây khung nhỏ nhất.
 - Đổi dấu so sánh, dấu bé ($<$) thành dấu lớn ($>$), trong các biểu thức so sánh của các thuật toán tìm cây khung.