CÂY KHUNG TRONG ĐỒ THỊ

Đặng Nguyễn Đức Tiến
Đặng Trần Minh Hậu
Nguyễn Ngọc Thảo

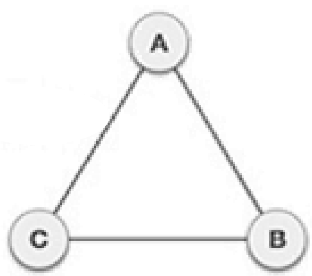
Nội dung bài giảng

- Khái niệm về cây trong lý thuyết đồ thị
- Định nghĩa cây khung
  - Xác định cây khung bằng DFS và BFS
- Cây khung nhỏ nhất
  - Xác định cây khung nhỏ nhất bằng giải thuật Prim và Kruskal

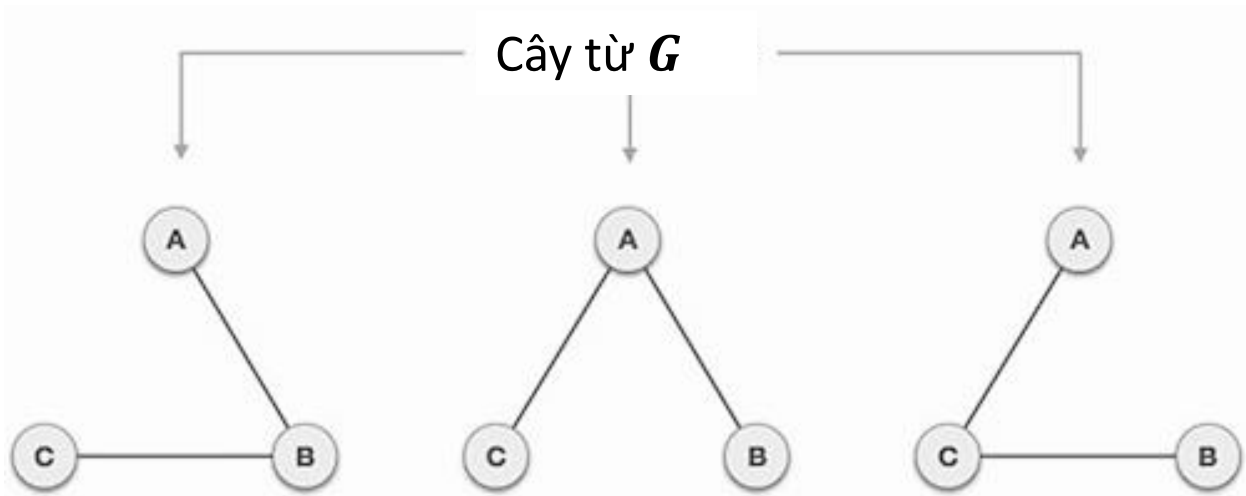
Khái niệm về cây

Khái niệm cây trong lý thuyết đồ thị

- Cây (tree) là một dạng đặc biệt của đồ thị vô hướng, liên thông, và không chứa chu trình.
  - Mọi cây là đồ thị, nhưng không phải đồ thị nào cũng là cây.
- Định nghĩa hàm ý rằng mọi cây không chứa khuyên cũng như cạnh song song vì các cạnh này sẽ tạo thành chu trình.

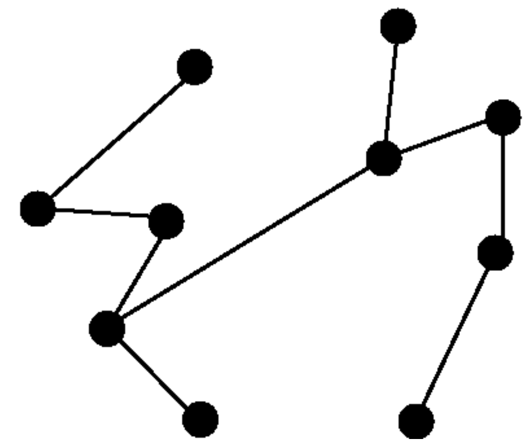


Đồ thị $\boldsymbol{G}$

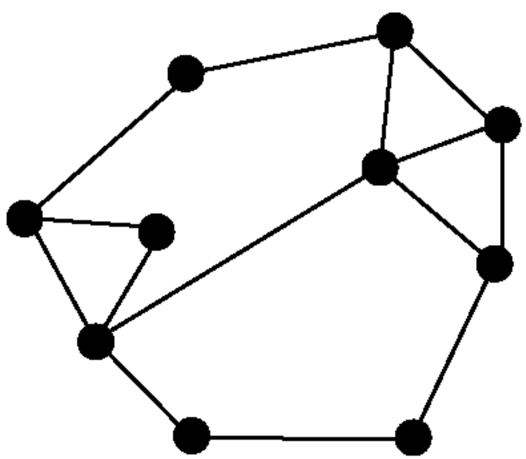


Khái niệm rừng trong lý thuyết đồ thị

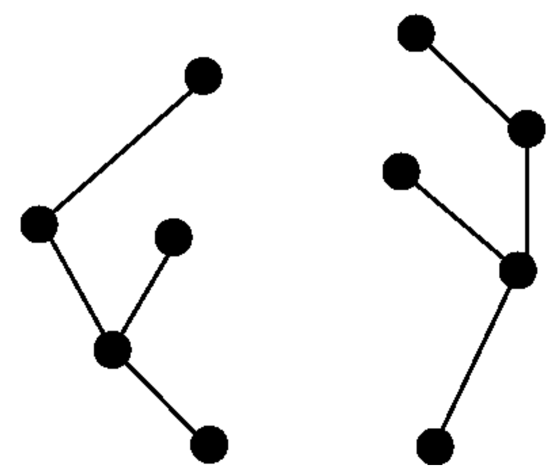
- Rừng (forest) là đồ thị mà mỗi thành phần liên thông là cây.
- Như vậy, đồ thị không có chu trình là một rừng.



Cây (không chu trình, liên thông)



Đồ thị (có chu trình)



Rừng (không chu trình, không liên thông)

Định lý về sự tồn tại các đỉnh treo

- Nếu cây $\boldsymbol{T}$ gồm $n \geq 2$ đỉnh thì $\boldsymbol{T}$ chứa ít nhất hai đỉnh treo.
- **Chứng minh**
  - Gọi $v_1, v_2, \dots, v_k$ là đường đi đơn dài nhất trên cây. Như vậy, v_1 và v_k là các đỉnh treo.
  - v_1 không có cạnh nối tới bất kỳ đỉnh nào trong các đỉnh $v_2, v_3, \dots, v_k$ do đồ thị không chứa chu trình, cũng như với bất cứ đỉnh nào của đồ thị do đường đi đang xét là dài nhất.
  - Do vậy, cây $\boldsymbol{T}$ phải chứa ít nhất hai đỉnh treo.

Các tính chất của cây

- Xét đơn đồ thị $\boldsymbol{T} = (\boldsymbol{V}, \boldsymbol{E})$ là đồ thị vô hướng n đỉnh.
- Khi đó, các mệnh đề sau đây là tương đương:

1. T là cây.
2. T không chứa chu trình và có $n - 1$ cạnh.
3. T liên thông và có $n - 1$ cạnh.
4. T liên thông và mỗi cạnh của nó đều là cầu.
5. Hai đỉnh bất kỳ của T được nối với nhau bởi đúng một đường đi đơn.
6. T không chứa chu trình nhưng khi thêm vào một cạnh bất kỳ ta sẽ thu được đúng một chu trình.

Chứng minh tính chất 1 → 2

- Nếu $\boldsymbol{T}$ là cây thì $\boldsymbol{T}$ không chứa chu trình và có $n - 1$ cạnh.
- **Chứng minh**
  - Với $n = 1$: T có 0 cạnh → đúng
  - Giả sử $n = k$ đúng, nghĩa là T có $k - 1$ cạnh.
  - Ta chứng minh cây T có $n = k + 1$ đỉnh sẽ có k cạnh.
  - Theo định lý về sự tồn tại các đỉnh treo, T có ít nhất hai đỉnh treo.
  - Gọi hai đỉnh treo là v_1 và v_2. Việc loại bỏ v_1 và cạnh nối với nó làm cho đồ thị còn lại là cây có k đỉnh, và theo giả thiết quy nạp, nó có $k - 1$ cạnh.
  - Vậy cây ban đầu có k cạnh. [đpcm]

Chứng minh tính chất 2 $\rightarrow$ 3

- Nếu $\boldsymbol{T}$ không chứa chu trình và có $n-1$ cạnh thì $\boldsymbol{T}$ liên thông và có $n-1$ cạnh.
- **Chứng minh**
  - Giả sử T không liên thông. Như vậy, T có $k \geq 2$ thành phần liên thông, $T_1, T_2, \dots, T_k$.
  - T không có chu trình nên các thành phần liên thông T_i cũng không có chu trình. Vậy, T_i là các cây.
  - Gọi $n(T_i)$ và $e(T_i)$ là số đỉnh và số cạnh của cây T_i. Do T_i là cây nên ta có: $e(T_i) = n(T_i) - 1$ (Tính chất 2)
  - Suy ra, $e(T) = e(T_1) + \cdots + e(T_k) = n(T_1) + \cdots + n(T_k) - k = n(T) - k$

    $= n - k < n - 1$ (mâu thuẫn vì $e(T) = n - 1$)
  - Do vậy, T liên thông. [đpcm]

Chứng minh tính chất 3 $\rightarrow$ 4

- Nếu $\boldsymbol{T}$ liên thông và có $n - 1$ cạnh thì mỗi cạnh của $\boldsymbol{T}$ là cầu và $\boldsymbol{T}$ liên thông.

- **Chứng minh**
  - Nếu loại bỏ một cạnh bất kỳ khỏi T thì số cạnh của T là $n - 2$.
  - T với n đỉnh và $n - 2$ cạnh chắc chắn không liên thông. Tức là, khi bỏ một cạnh bất kỳ trong T, số thành phần liên thông tăng thêm.
  - Do đó, theo định nghĩa cạnh cầu, ta có mọi cạnh trong T đều là cạnh cầu. [đpcm]

Chứng minh tính chất 4 $\rightarrow$ 5

- Nếu T liên thông và mỗi cạnh đều là cầu thì hai đỉnh bất kỳ của T được nối với nhau bởi đúng một đường đi.

- **Chứng minh**
  - Do T liên thông, nên hai đỉnh bất kỳ của nó đều có thể nối với nhau bởi một đường đi đơn.
  - Nếu có cặp đỉnh nào của T có hai đường đi đơn khác nhau nối chúng thì uy ra đồ thị chứa chu trình. Như vậy, các cạnh trên chu trình này không phải là cầu.
  - Điều này mâu thuẫn vì Tính chất 4 cho biết mọi cạnh đều là cầu.

Chứng minh tính chất 5 $\rightarrow$ 6

- Nếu hai đỉnh bất kỳ của $\boldsymbol{T}$ được nối với nhau bởi đúng một đường đi đơn thì $\boldsymbol{T}$ không chứa chu trình, nhưng khi thêm vào một cạnh ta thu được đúng một chu trình.
- **Chứng minh**
  - T không chứa chu trình, vì nếu có, ta sẽ tìm được hai đỉnh có hai đường đi đơn phân biệt nối với nhau.
  - Nếu thêm cạnh (u, v) bất kỳ thì cạnh này và đường đi đơn từ u đến v hiện có ạo thành chu trình.
  - Chu trình này là duy nhất vì nếu thu được nhiều hơn một chu trình thì chứng tỏ rằng trước đó trong T đã có chu trình.

Chứng minh tính chất 6 $\rightarrow$ 1

- Nếu $\boldsymbol{T}$ không chứa chu trình, nhưng hễ cứ thêm một cạnh bất kỳ vào $\boldsymbol{T}$ thì ta thu được đúng một chu trình thì $\boldsymbol{T}$ là cây ($\boldsymbol{T}$ liên thông và không có chu trình).

- **Chứng minh**
  - Giả sử T không liên thông. Như vậy, T có ít nhất hai thành phần liên thông.
  - Nếu thêm vào T một cạnh nối hai thành phần liên thông bất kỳ với nhau ta sẽ không thêm được chu trình. Mẫu thuẫn với tính chất 6.
  - Do vậy, T liên thông.

Bài tập rèn luyện

- Vẽ tất cả các cây không đẳng cấu có 4 đỉnh, 5 đỉnh, và 6 đỉnh.

Bài tập rèn luyện

Có thể tìm được một cây có 8 đỉnh và thỏa các điều kiện dưới đây hay không? Nếu có, hãy vẽ cây đó ra, nếu không, hãy giải thích.

a) Mọi đỉnh đều có bậc 1

b) Mọi đỉnh đều có bậc 2

c) Có 6 đỉnh bậc 2 và 2 đỉnh bậc 1

d) Có 1 đỉnh bậc 7 và 7 đỉnh bậc 1

Bài tập rèn luyện

Các phát biểu sau đây đúng hay sai? Giải thích.

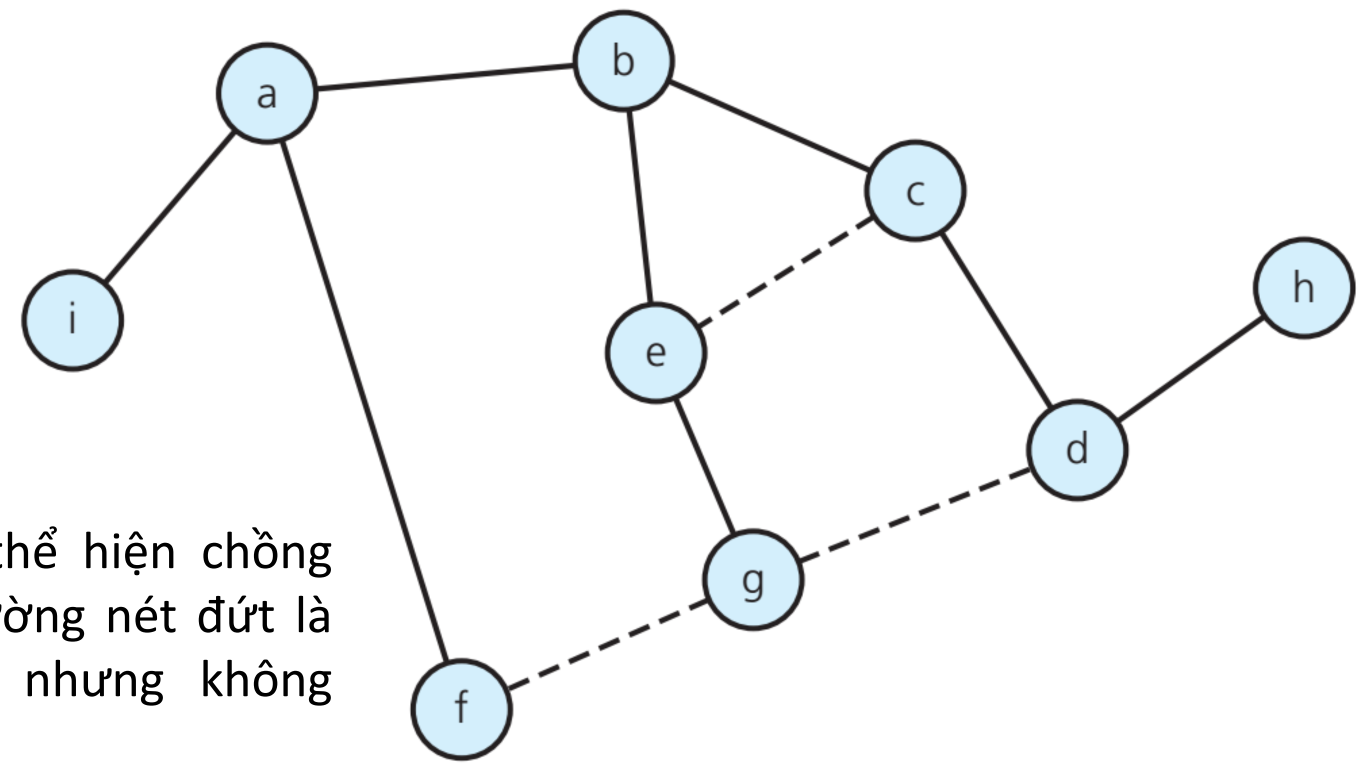
Lưu ý: Ta xét trên đồ thị đơn vô hướng.

1. Nếu đồ thị có n đỉnh và $n - 1$ cạnh thì G là một cây.
2. Nếu G liên thông thì G không có chu trình.
3. Nếu hủy một cạnh bất kỳ của đồ thị liên thông G cũng làm mất tính liên thông của G thì G không có chu trình.
4. Thêm một cạnh vào một cây sẽ sinh ra đúng một chu trình.
5. Nếu G không có chu trình và có 25 cạnh, 26 đỉnh thì G liên thông.
6. Nếu G có 32 cạnh, 28 đỉnh thì G không là cây.
7. Nếu G liên thông, có 10 cạnh, 10 đỉnh thì G có ít nhất một chu trình.

Cây khung

Định nghĩa cây khung

- Nếu đồ thị bộ phận $\boldsymbol{T}$ của đồ thị vô hướng liên thông $\boldsymbol{G} = (\boldsymbol{V}, \boldsymbol{E})$ là cây thì $\boldsymbol{T}$ là cây khung (spanning tree) của $\boldsymbol{G}$.



Cây khung được thể hiện chồng lên đồ thị gốc. Đường nét đứt là cạnh của đồ thị nhưng không thuộc cậy 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.
  - T là cây nên có đường đi trên T giữa hai 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 hai đỉnh bất kỳ.
  - Do đó, G liên thông.

Định lý về tính liên thông

- **Chứng minh chiều thuận**
  - Giả sử G liên thông.
  - Nếu G không là cây thì nó phải có một chu trình đơn. Nếu xóa một cạnh bất kỳ trong chu trình đơn thì đồ thị bộ phận vẫn liên thông.
  - Nếu đồ thị bộ phận này không là cây thì nó còn chứa chu trình. Ta 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.
  - Đồ thị được tạo ra vẫn còn liên thông. Đồ thị này là cây khung vì là đồ thị bộ phận của G.

Xác định cây khung

- Cây khung có thể được xác định bằng giải thuật duyệt đồ thị theo chiều sâu (DFS) hoặc duyệt theo chiều rộng (BFS).
- Quá trình duyệt các đỉnh chính là quá trình tìm cây khung của đồ thị.
- Số cây khung của đồ thị đầy đủ K_n là n^{n-2} (định lý Borchart, Sylvester, Cayley)

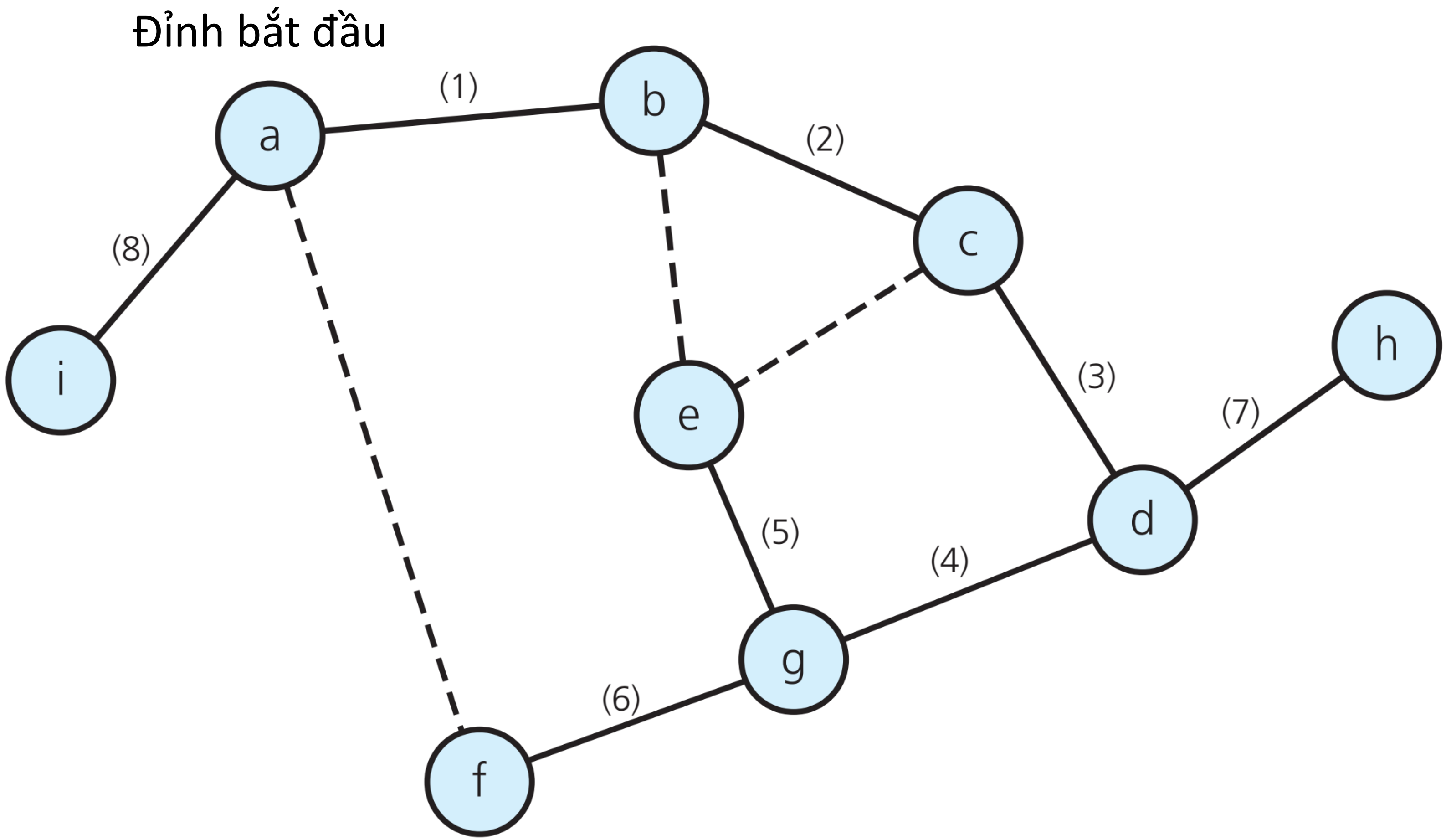
Bài tập rèn luyện

1. Giả sử đồ thị G liên thông, có 13 đỉnh và 20 cạnh. Cây khung của G có bao nhiêu đỉnh? Có bao nhiêu cạnh?
2. Đồ thị đầy đủ K_5 có bao nhiêu cây khung?

Xác định cây khung bằng DFS

- Ta bổ sung một số hiệu chỉnh đơn giản vào hàm duyệt đồ thị theo chiều sâu DFS.

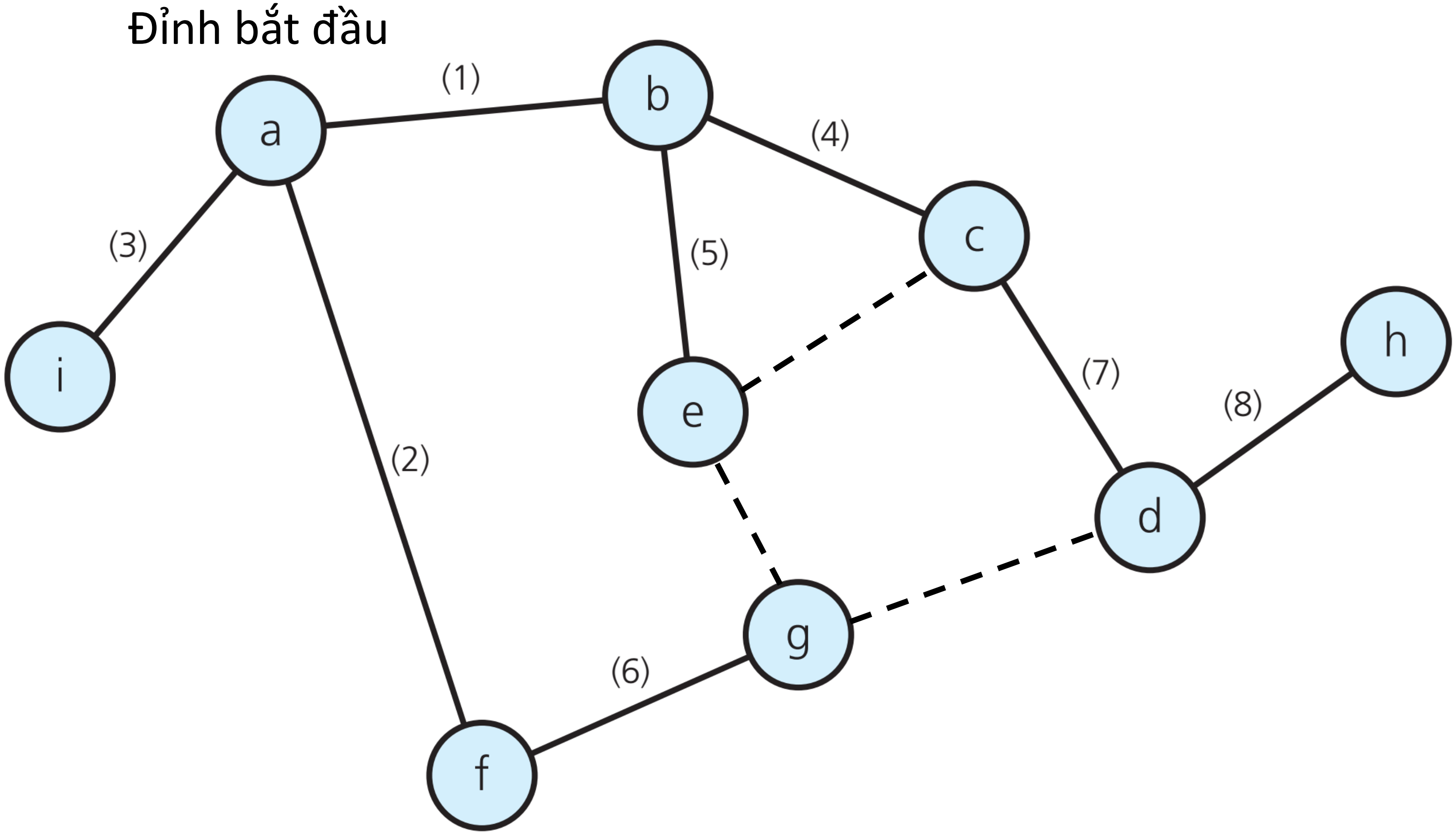
```
// Dựng cây khung cho đồ thị vô hướng liên thông tại đỉnh v
// bằng duyệt đồ thị theo chiều sâu
dfsTree(v: Đỉnh)
    Đánh dấu v đã viếng thăm
    for (mỗi đỉnh chưa viếng thăm u kề với v){
        Đánh dấu cạnh từ u tới v
        dfsTree(u)
    }
```



Giải thuật DFS phát triển cây khung từ đỉnh a bằng cách viếng thăm các đỉnh theo thứ tự: a, b, c, d, g, e, f, h, i.
Các số thể hiện thứ tự đánh dấu cạnh do DFS thực hiện.

Xác định cây khung bằng BFS

```
// Dựng cây khung cho đồ thị vô hướng liên thông tại đỉnh v bằng BFS
bfsTree(v: Đỉnh)
    q = Hàng đợi mới khởi tạo rỗng
    q.enqueue(v)                // đưa v vào trong hàng đợi
    Đánh dấu v đã viếng thăm
    while (!q.isEmpty()){
        q.dequeue(w)            // lấy đỉnh w ở đầu hàng đợi ra
        // Tồn tại đường đi từ đỉnh w đến mọi đỉnh trong trong q
        for (mỗi đỉnh chưa viếng thăm u kề với w){
            Đánh dấu u đã viếng thăm
            Đánh dấu cạnh giữa w và u
            q.enqueue(u)
        }
    }
```



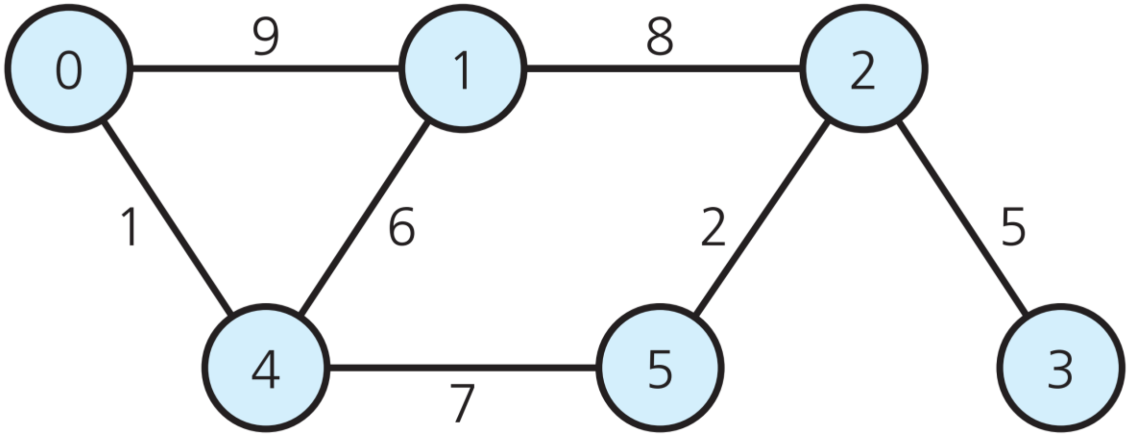
Giải thuật BFS phát triển cây khung từ đỉnh a bằng cách viếng thăm các đỉnh theo thứ tự: a, b, f, i, c, e, g, d, h.

Các số thể hiện thứ tự đánh dấu cạnh do DFS thực hiện.

Bài tập rèn luyện

Cho đồ thị như bên dưới.

Giả sử đỉnh bắt đầu là 0. Hãy xác định cây khung lần lượt theo giải thuật duyệt đồ thị DFS và BFS.

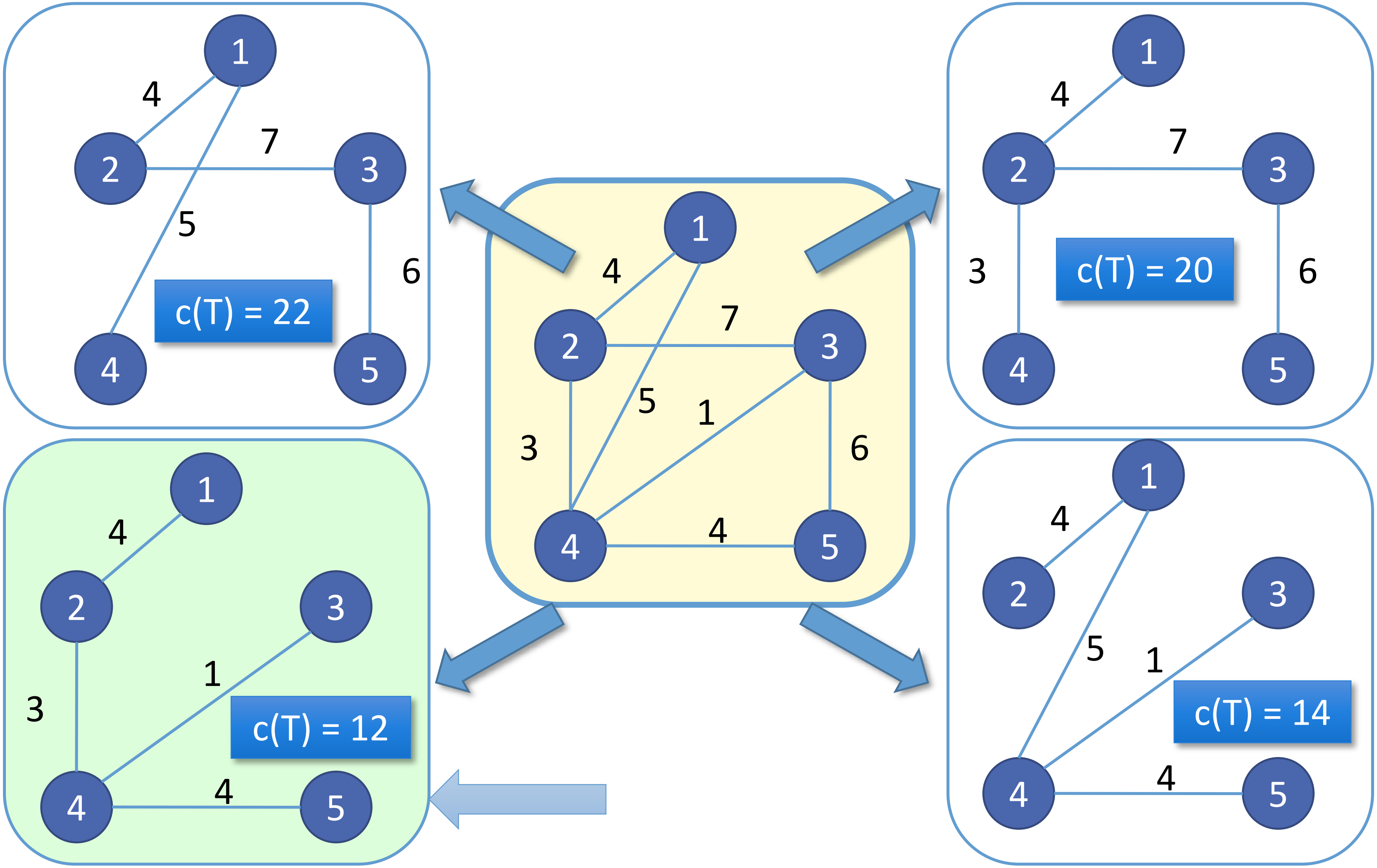


Cây khung nhỏ nhất

Cây khung nhỏ nhất (MST)

- Chi phí của cây khung $\boldsymbol{T}$ là tổng chi phí của các cạnh thuộc $\boldsymbol{T}$, ký hiệu $c(\boldsymbol{T})$.
- Cây khung nhỏ nhất (minimum spanning tree, MST) là cây khung có chi phí tối thiểu.
  - Có thể có nhiều cây khung nhỏ nhất nếu đồ thị chứa nhiều cạnh cùng trọng số.

Cây khung lớn nhất

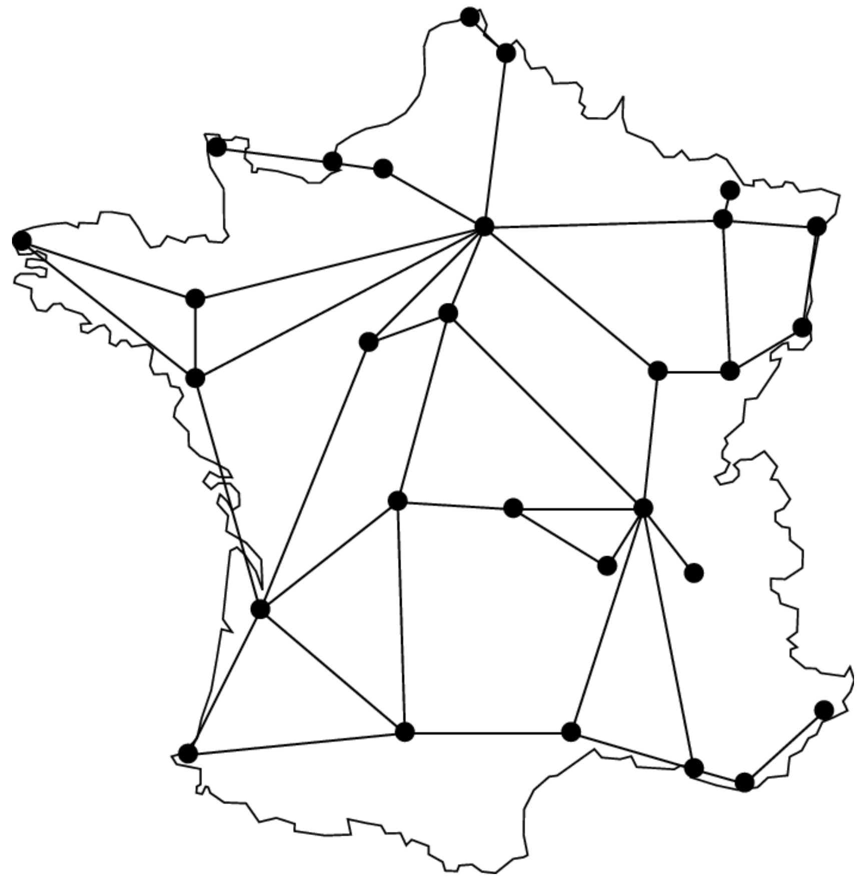


Cây khung nhỏ nhất

Cây khung nhỏ nhất: Ứng dụng

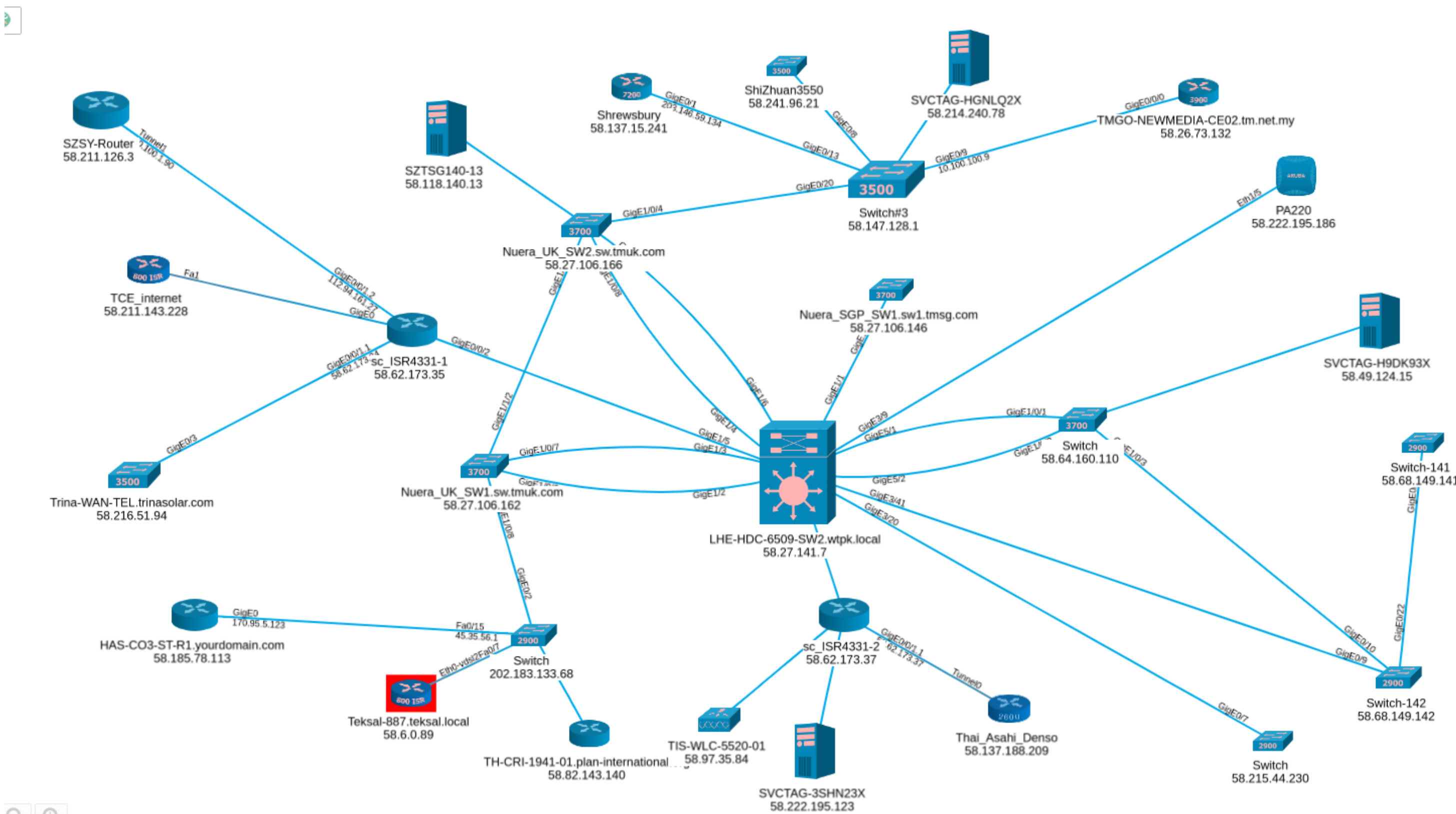
- Bài toán xây dựng hệ thống đường cao tốc

Xây dựng 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, vì lý do kinh tế, chi phí xây dựng hệ thống đường phải là nhỏ nhất.



Cây khung nhỏ nhất: Ứng dụng

- Bài toán nối mạng máy tính: Kết nối hệ thống máy tính gồm N thiết bị, biết rằng chi phí nối thiết bị i với thiết bị j là $c[i,j]$, sao cho tổng chi phí nối mạng là nhỏ nhất.



Giải thuật tìm cây khung nhỏ nhất

- Prim và Kruskal là hai giải thuật thông dụng nhất để tìm cây khung nhỏ nhất.
- Giải thuật Prim do Robert Prim đưa ra vào năm 1957
- Giải thuật Kruskal do Joseph Kruskal đề xuất vào năm 1956.

Robert Prim
(1921 – 2021)

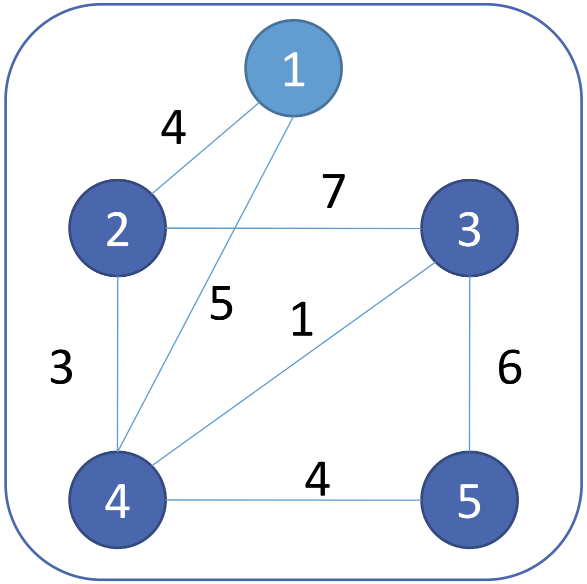
Joseph Kruskal
(1928 – 2010)

Giải thuật Prim

- Cho $\boldsymbol{G} = (\boldsymbol{V}, \boldsymbol{E})$ là đồ thị liên thông có trọng số gồm n đỉnh.
- Bước 1. Chọn một đỉnh bất kỳ $u \in \boldsymbol{V}$ và khởi tạo $Y = \{u\}$ và $\boldsymbol{E_T} = \emptyset$.
- Bước 2. Trong các cạnh $e = \{u, v\}$ sao cho $u \in \boldsymbol{Y}$ và $v \in \boldsymbol{V} \backslash \boldsymbol{Y}$, chọn cạnh $e_i = \{u_i, v_i\}$ có độ dài nhỏ nhất.
- Bước 3. Gán $\boldsymbol{Y} = \boldsymbol{Y} \cup \{v_i\}$ và $\boldsymbol{E_T} = \boldsymbol{E_T} \cup \{e_i\}$.
- Bước 4. Nếu $\boldsymbol{E_T}$ đủ $n - 1$ phần tử thì dừng, ngược lại làm tiếp Bước 2.
- $\boldsymbol{T} = (\boldsymbol{V}, \boldsymbol{E_T})$ là cây khung nhỏ nhất.

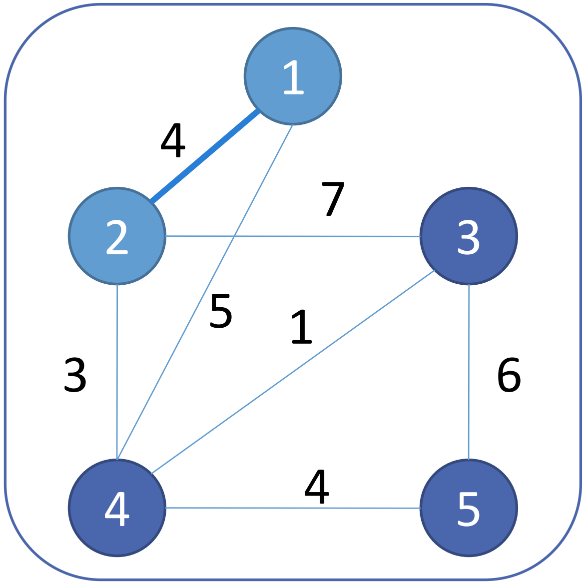
Ví dụ minh họa giải thuật Prim

- Giả sử chọn đỉnh 1 là đỉnh bắt đầu
- Cập nhật
  - $\boldsymbol{Y} = \{1\}$
  - $\boldsymbol{E_T} = \varnothing$



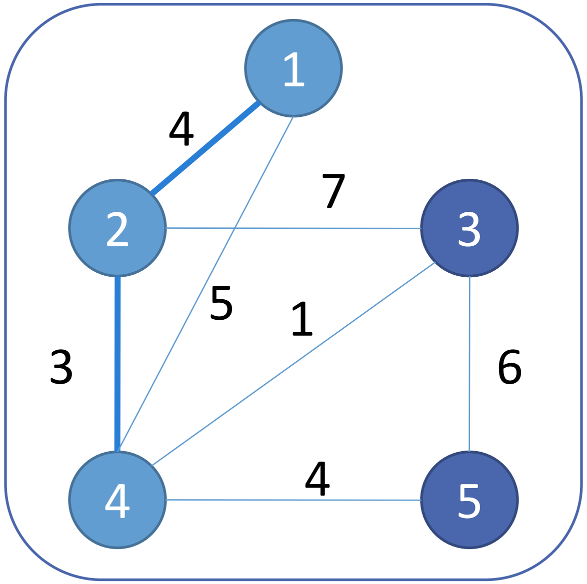
Ví dụ minh họa giải thuật Prim

- 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:
  - $\boldsymbol{Y} = \{1, 2\}$
  - $\boldsymbol{E_T} = \{\{1, 2\}\}$



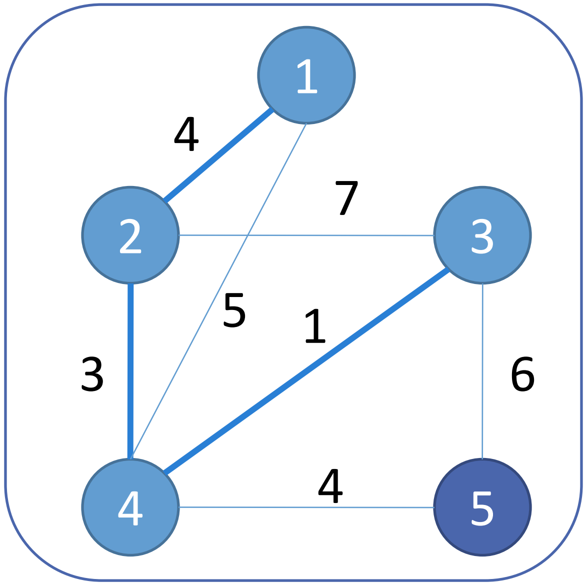
Ví dụ minh họa giải thuật Prim

- 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:
  - $\boldsymbol{Y} = \{1, 2, 4\}$
  - $\boldsymbol{E_T} = \{\{1, 2\}, \{2, 4\}\}$



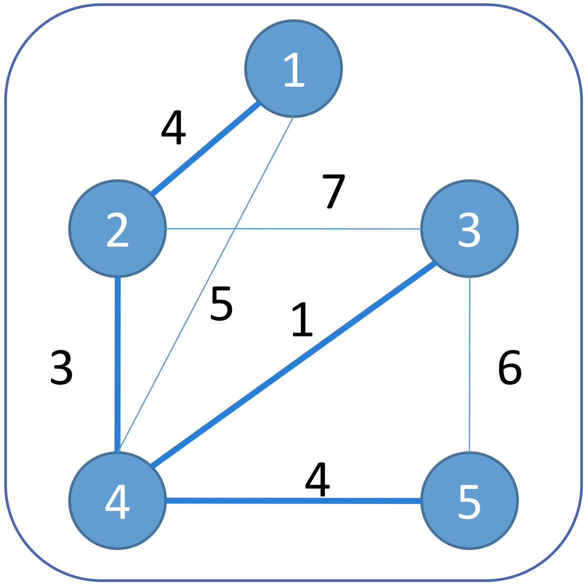
Ví dụ minh họa giải thuật Prim

- 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:
  - $\boldsymbol{Y} = \{1, 2, 4, 3\}$
  - $\boldsymbol{E_T} = \{\{1, 2\}, \{2, 4\}, \{4, 3\}\}$



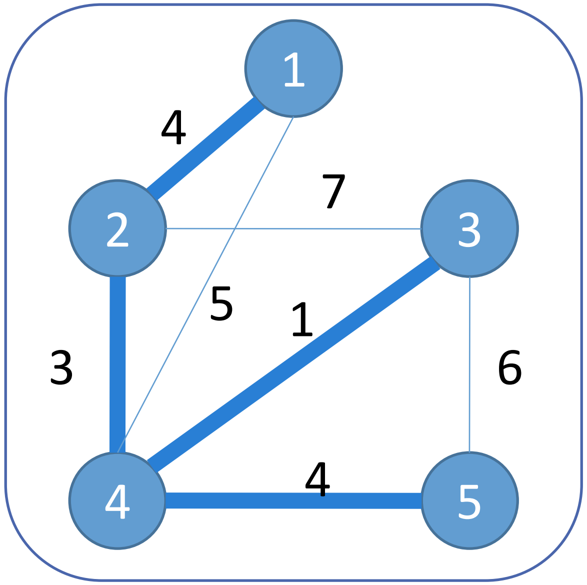
Ví dụ minh họa giải thuật Prim

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



Ví dụ minh họa giải thuật Prim

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

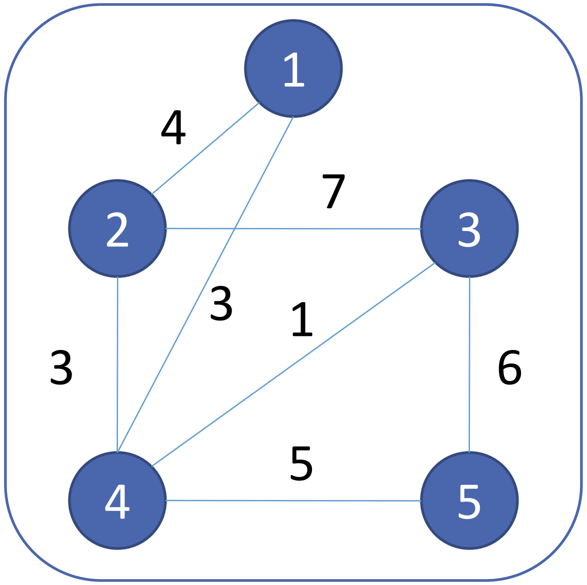


Giải thuật Kruskal

- Cho $\boldsymbol{G} = (\boldsymbol{V}, \boldsymbol{E})$ là 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: $\boldsymbol{E_T} = \emptyset$.
- Bước 2. Lần lượt lấy từng cạnh e trong danh sách đã sắp xếp. Nếu $\boldsymbol{E_T} \cup \{e\}$ không tạo thành chu trình trong T thì gán $\boldsymbol{E_T} = \boldsymbol{E_T} \cup \{e\}$.
- Bước 3. Nếu $\boldsymbol{E_T}$ đủ $n - 1$ phần tử thì dừng, ngược lại làm tiếp Bước 2.
- $\boldsymbol{T} = (\boldsymbol{V}, \boldsymbol{E_T})$ là cây khung nhỏ nhất.

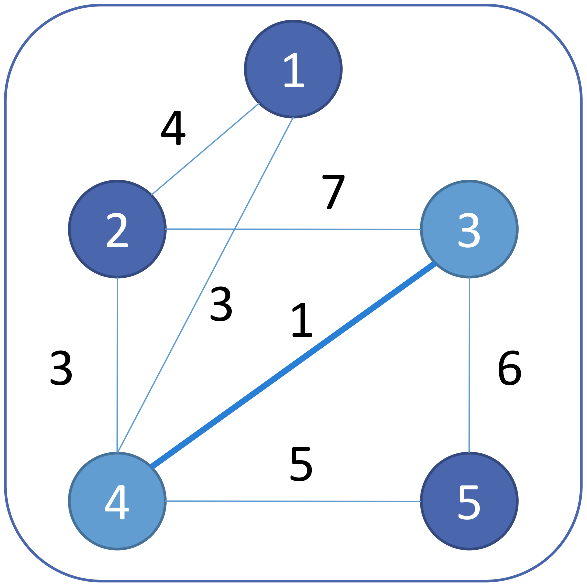
Ví dụ minh họa giải thuật Kruskal

- Sắp xếp các cạnh theo thứ tự tăng dần
  - $\{3,4\} = 1$
  - $\{1,4\} = 3$
  - $\{2,4\} = 3$
  - $\{1,2\} = 4$
  - $\{4,5\} = 5$
  - $\{3,5\} = 6$
  - $\{2,3\} = 7$
- $\boldsymbol{E_T} = \emptyset$



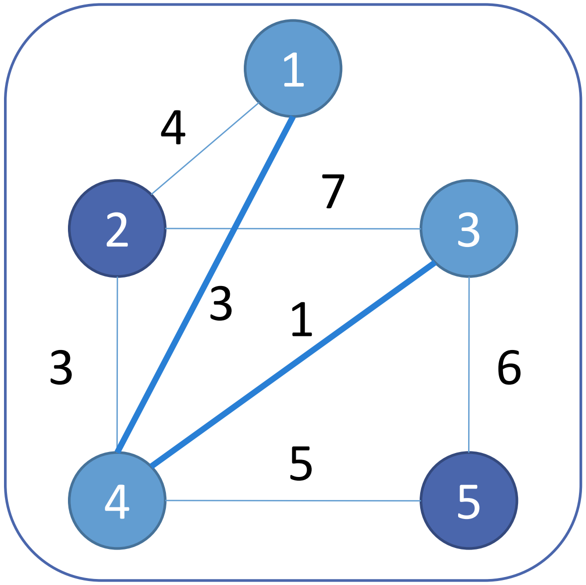
Ví dụ minh họa giải thuật Kruskal

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



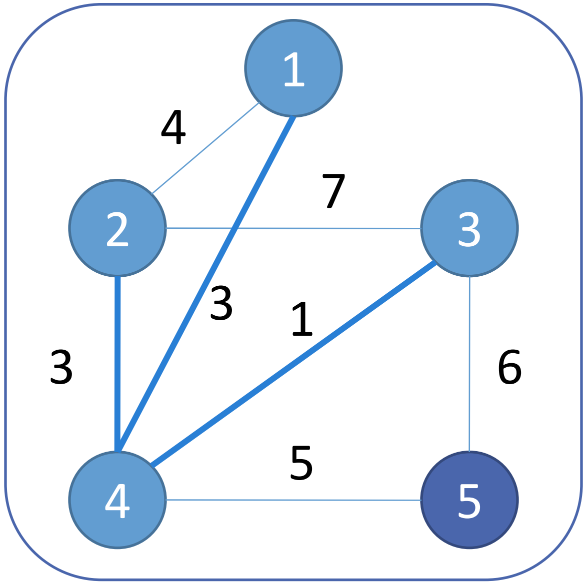
Ví dụ minh họa giải thuật Kruskal

- Chọn cạnh {1, 4} – là cạnh tiếp theo trong danh sách.
- $\boldsymbol{E_T} = \{\{3,4\},\{1,4\}\}$
- 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



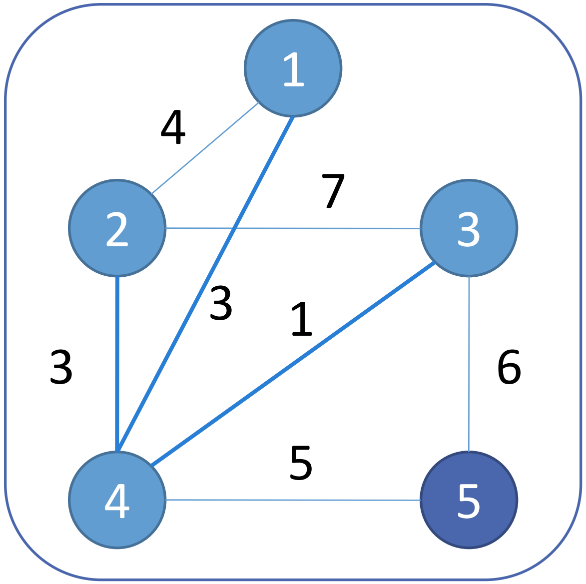
Ví dụ minh họa giải thuật Kruskal

- Chọn cạnh {2, 4} – là cạnh tiếp theo trong danh sách.
- $\boldsymbol{E_T} = \{\{3,4\},\{1,4\},\{2,4\}\}$
- 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



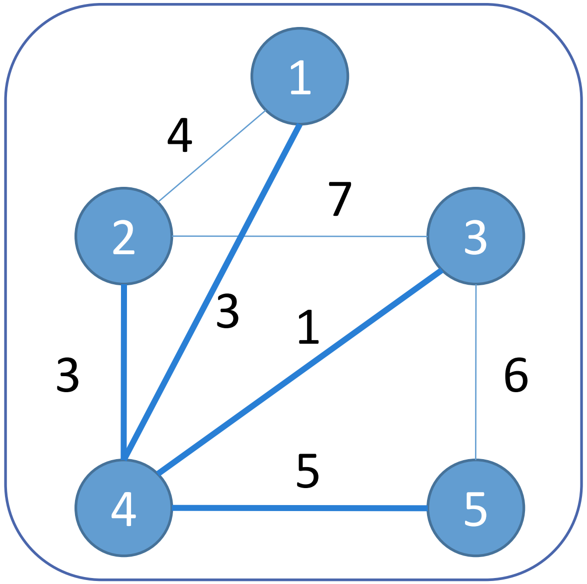
Ví dụ minh họa giải thuật Kruskal

- KHÔNG chọn cạnh {1, 2} – là cạnh tiếp theo trong danh sách vì khi chọn sẽ tạo thành chu trình.
- $\boldsymbol{E_T} = \{\{3,4\},\{1,4\},\{2,4\}\}$
- 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



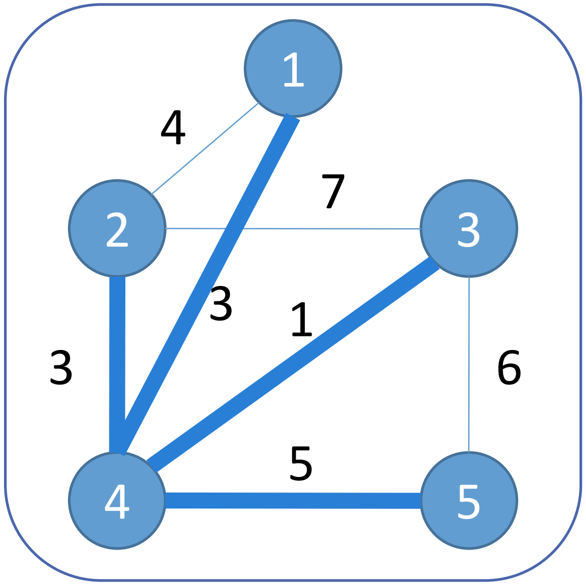
Ví dụ minh họa giải thuật Kruskal

- Chọn cạnh {4, 5} – là cạnh tiếp theo trong danh sách.
- $\boldsymbol{E_T} = \{\{3,4\},\{1,4\},\{2,4\},\{4,5\}\}$
- 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 minh họa giải thuật Kruskal

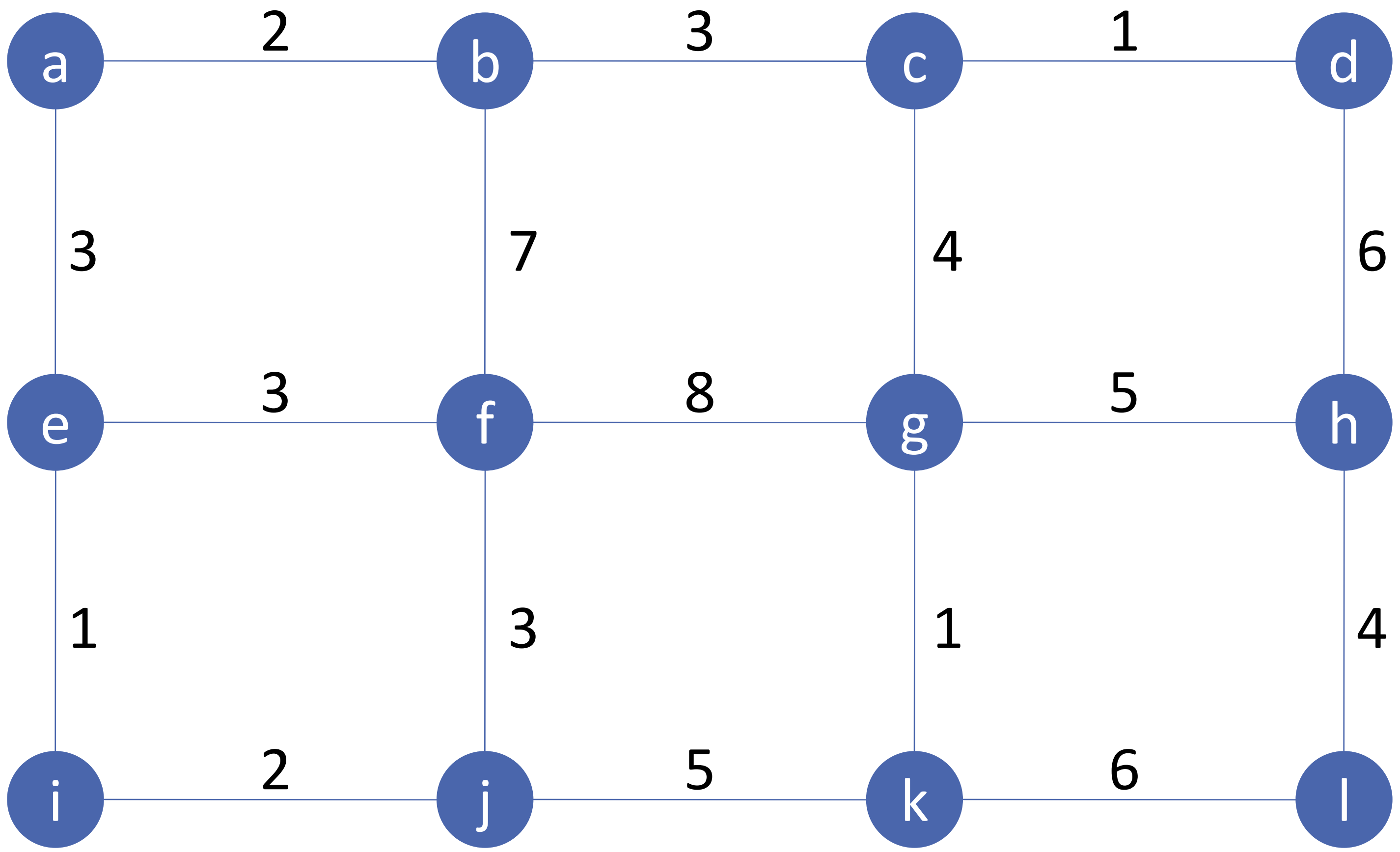
- Do $|\boldsymbol{E_T}| = 4$, kết thúc giải thuật.
- Cây khung tìm được là: $\boldsymbol{E_T} = \{\{3,4\},\{1,4\},\{2,4\},\{4,5\}\}$



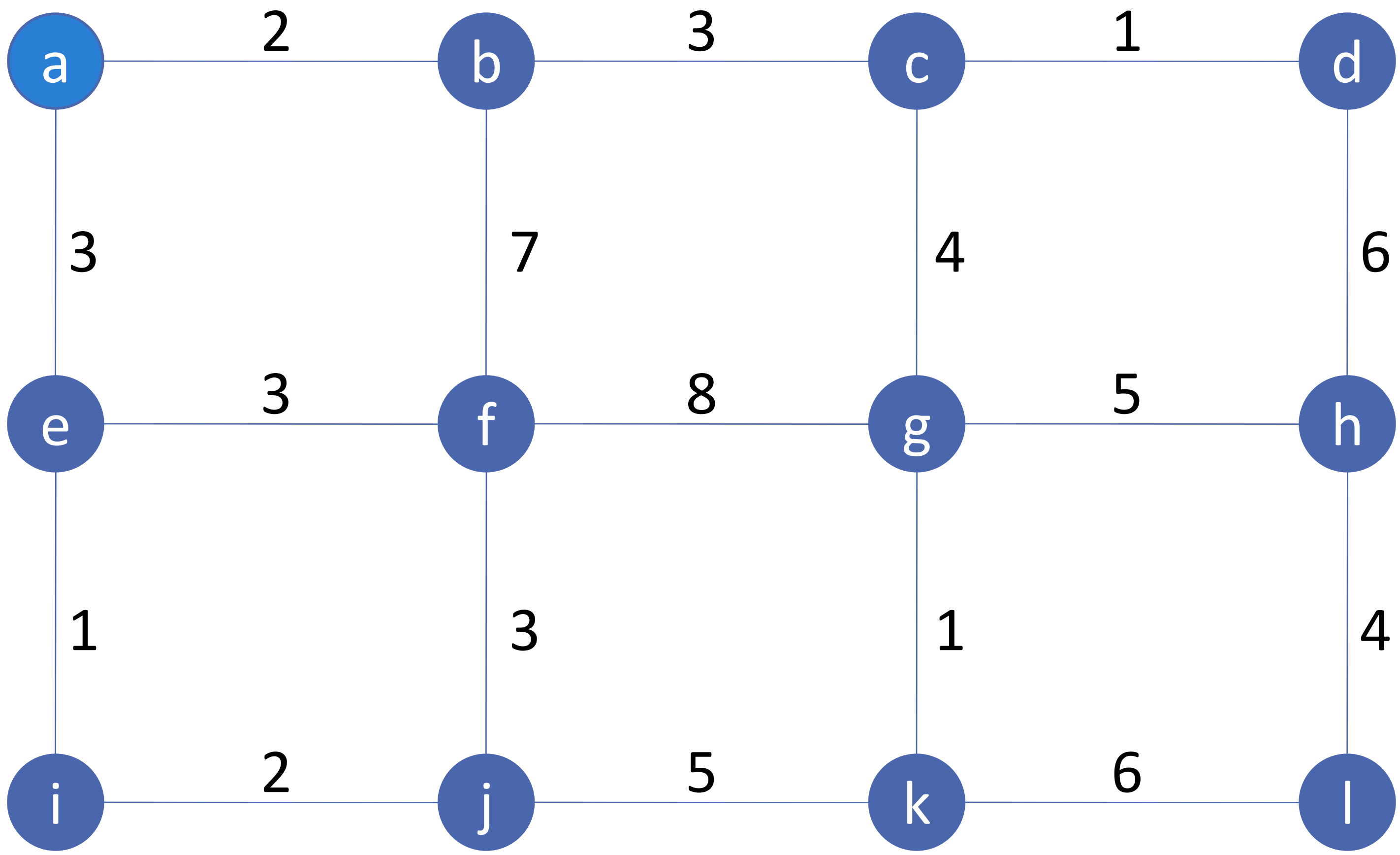
Xác định cây khung lớn nhất

- 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 giải thuật tìm cây khung.

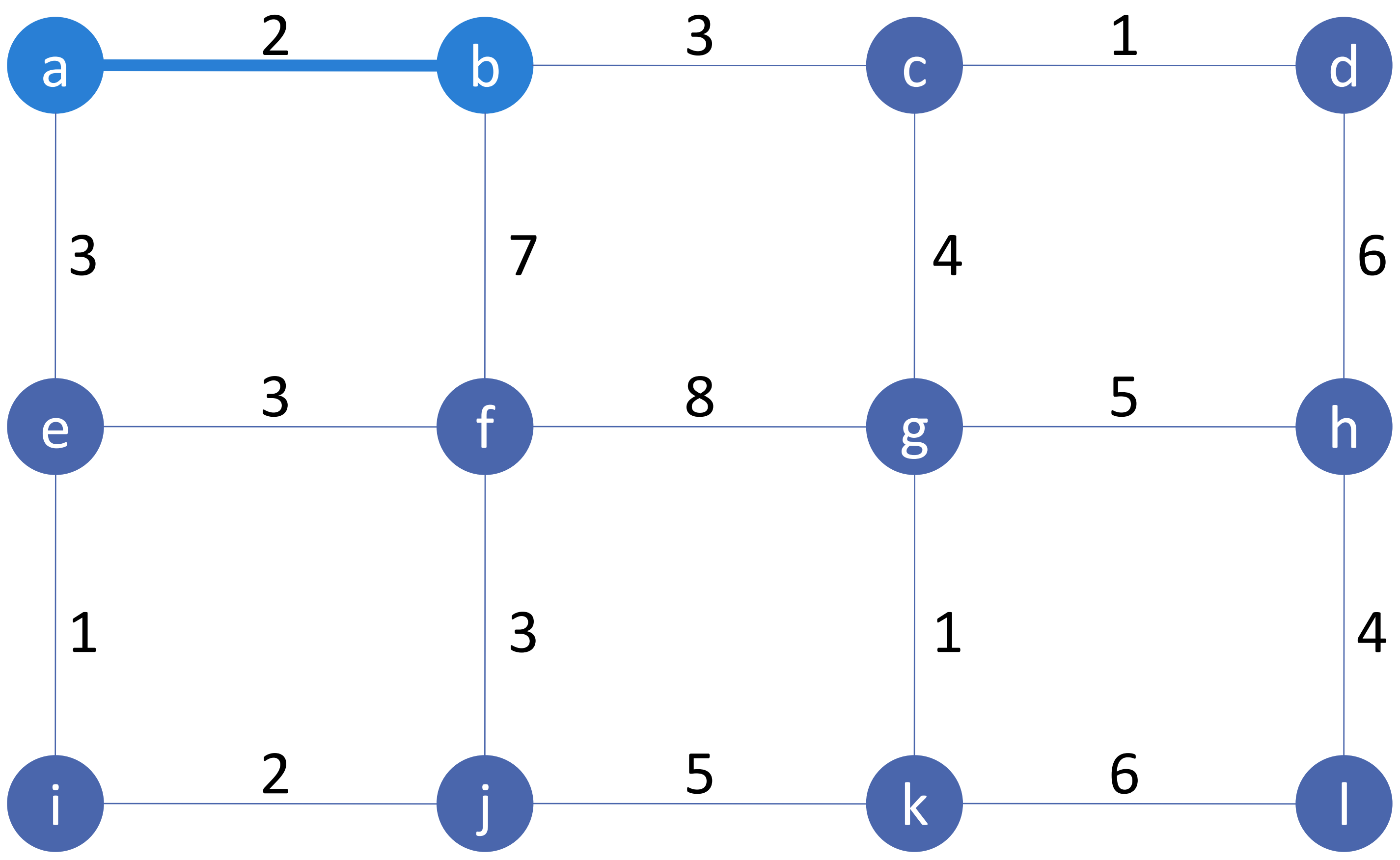
Ví dụ khác về giải thuật Prim



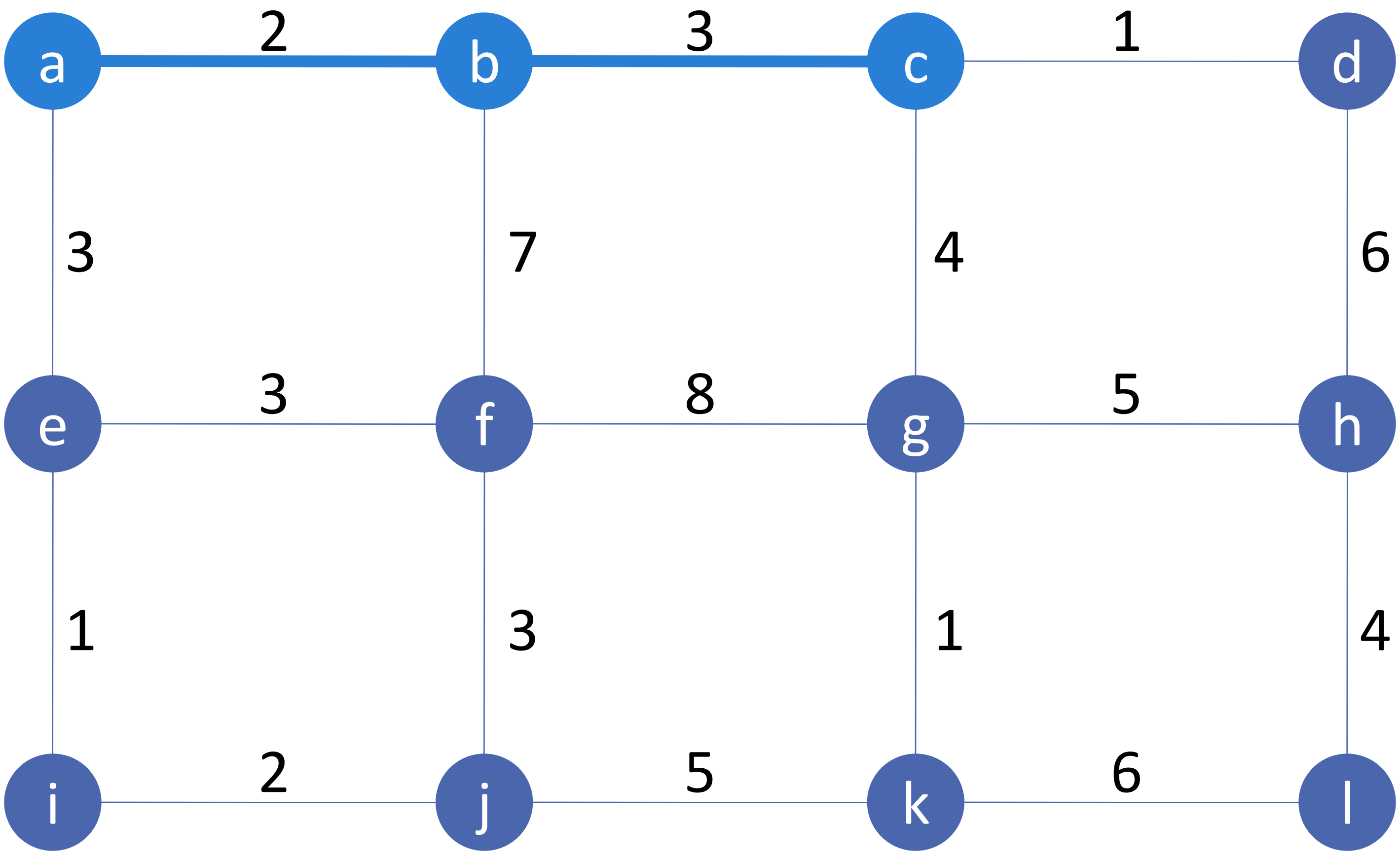
Ví dụ khác về giải thuật Prim



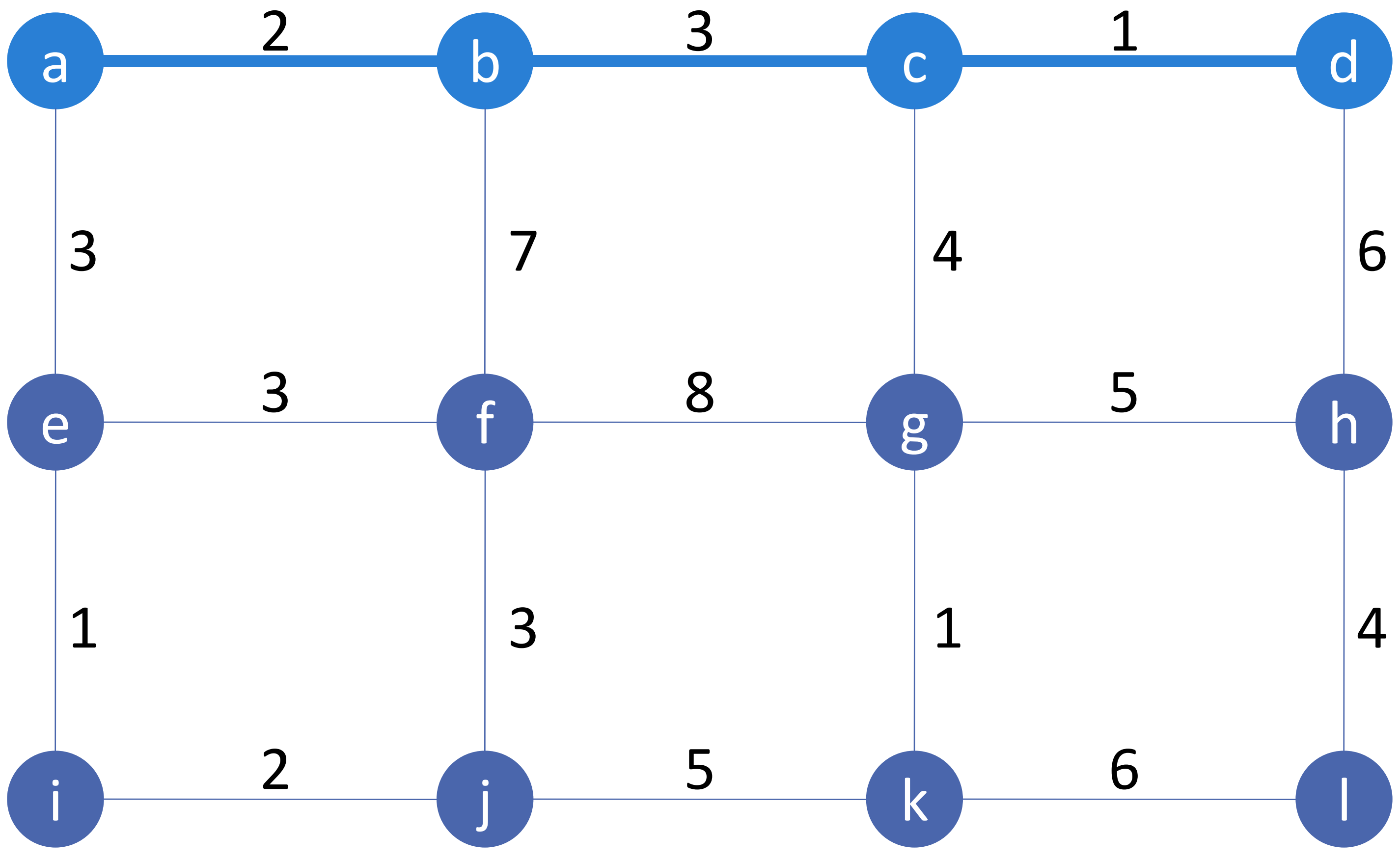
Ví dụ khác về giải thuật Prim



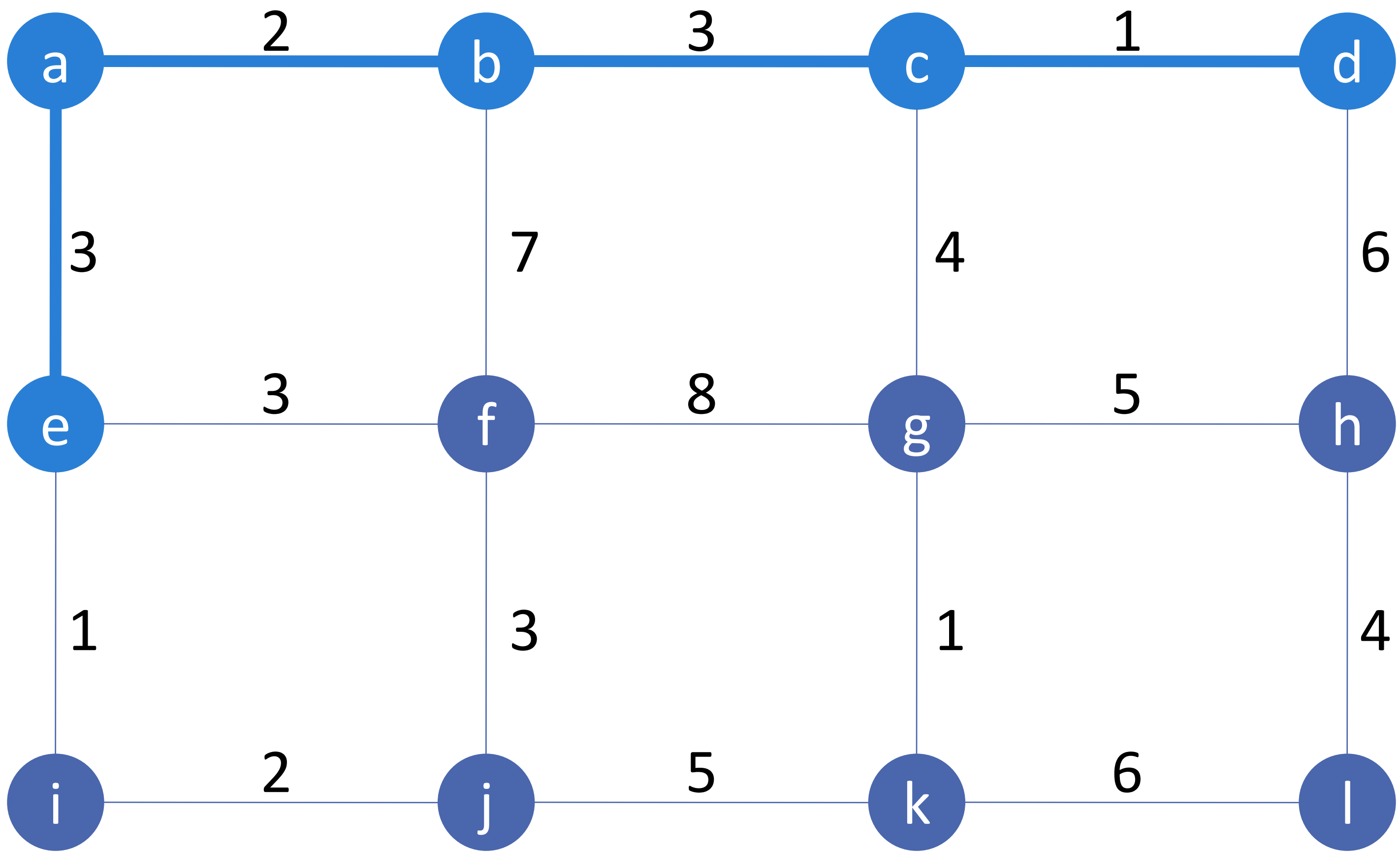
Ví dụ khác về giải thuật Prim



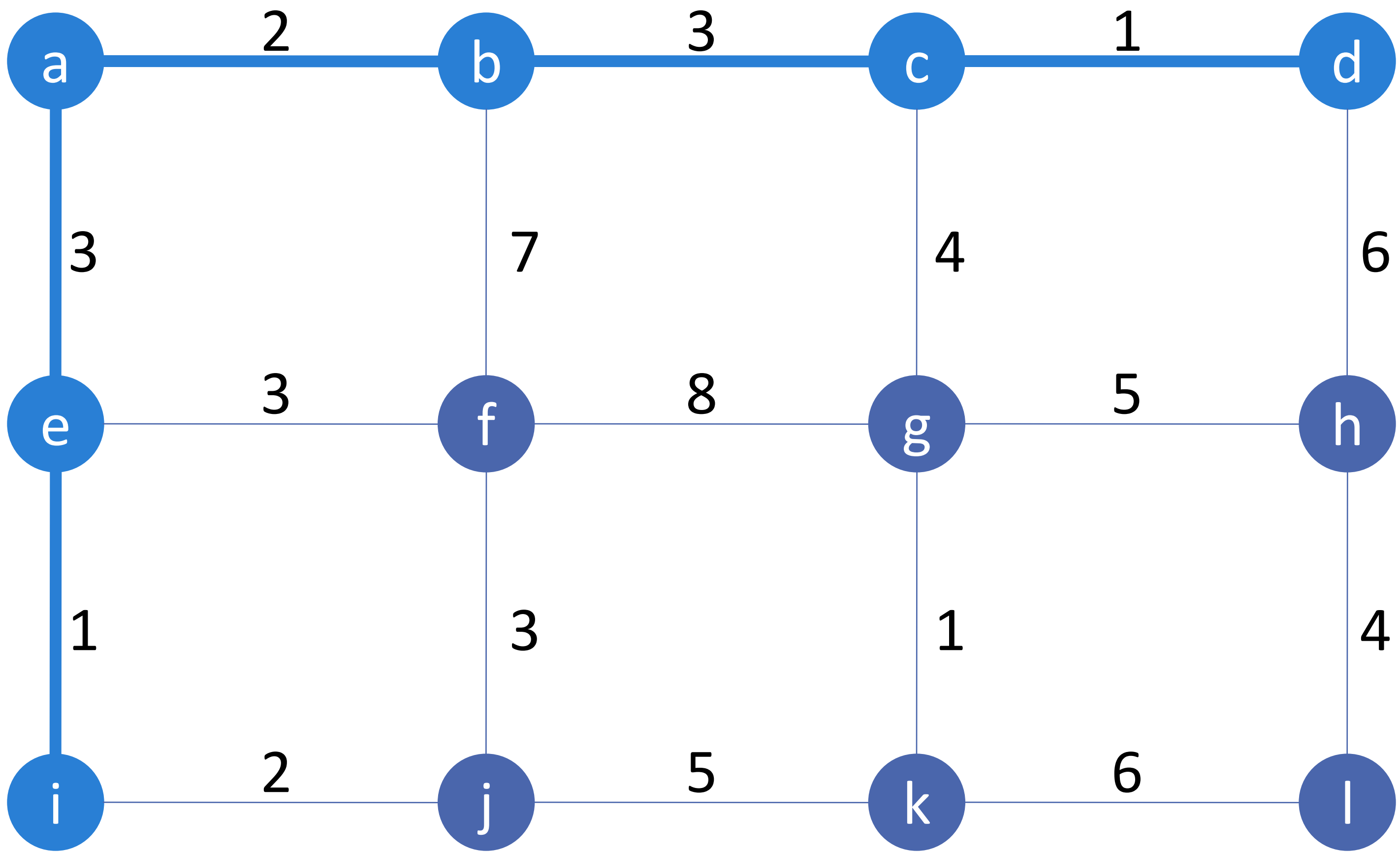
Ví dụ khác về giải thuật Prim



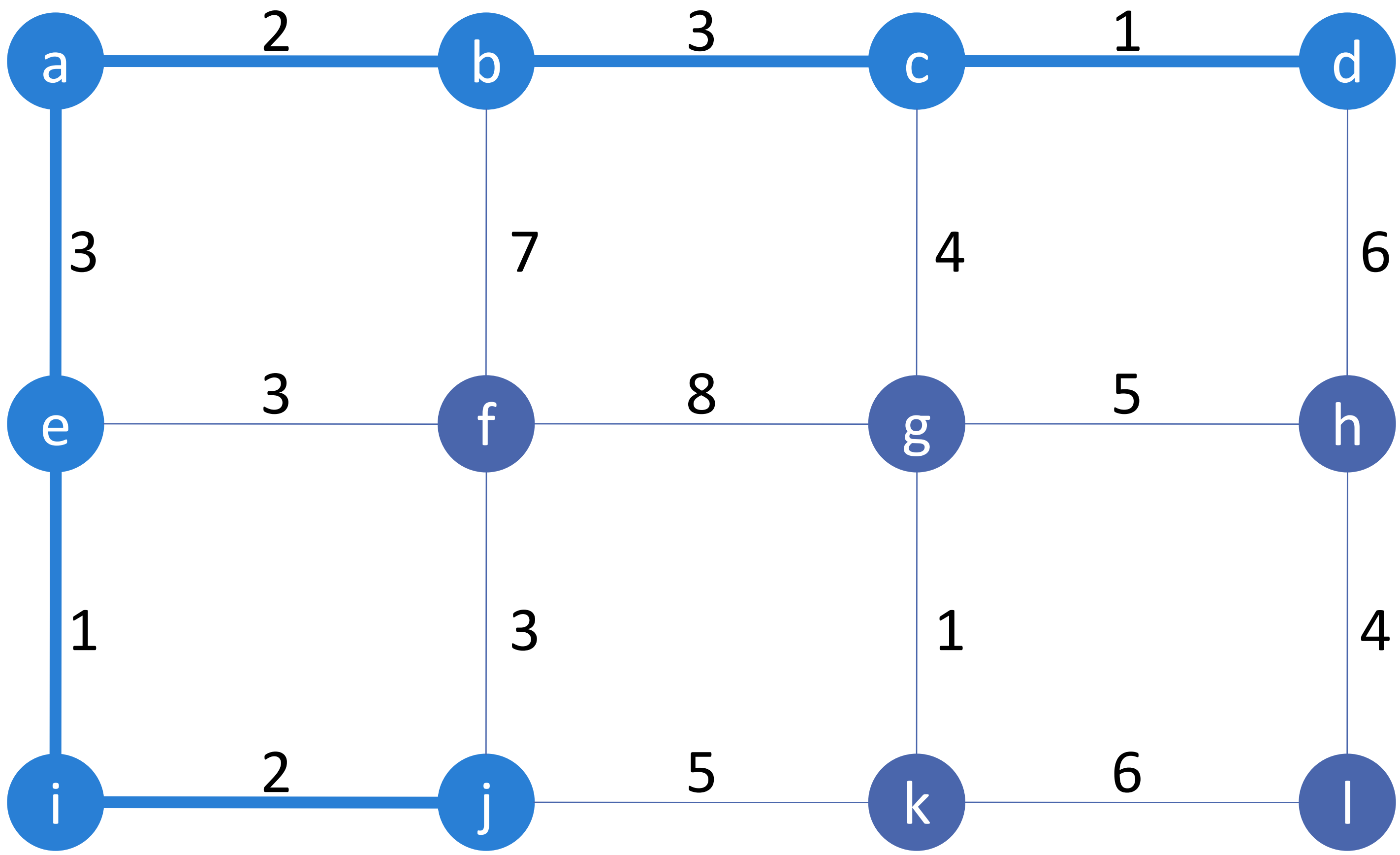
Ví dụ khác về giải thuật Prim



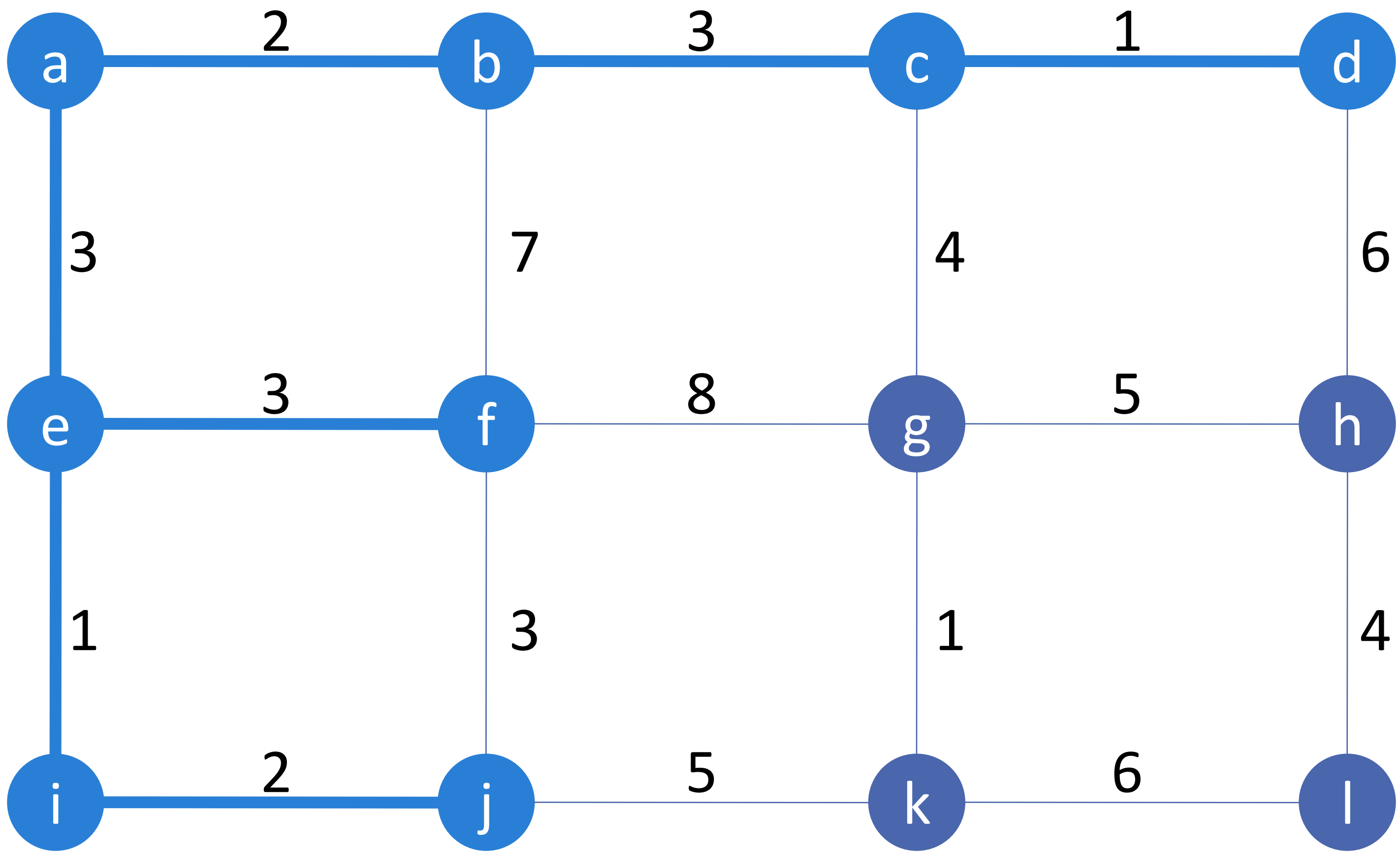
Ví dụ khác về giải thuật Prim



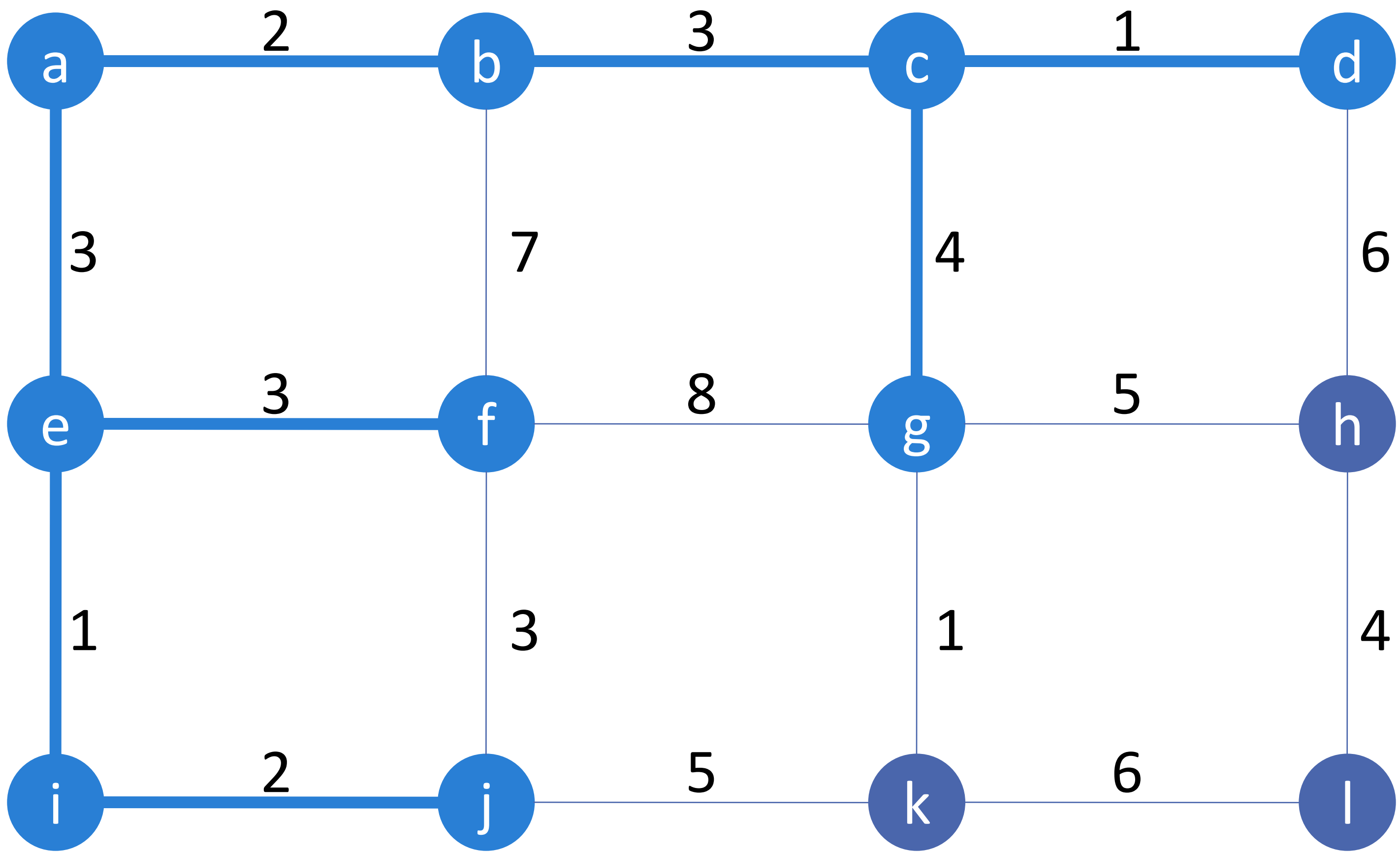
Ví dụ khác về giải thuật Prim



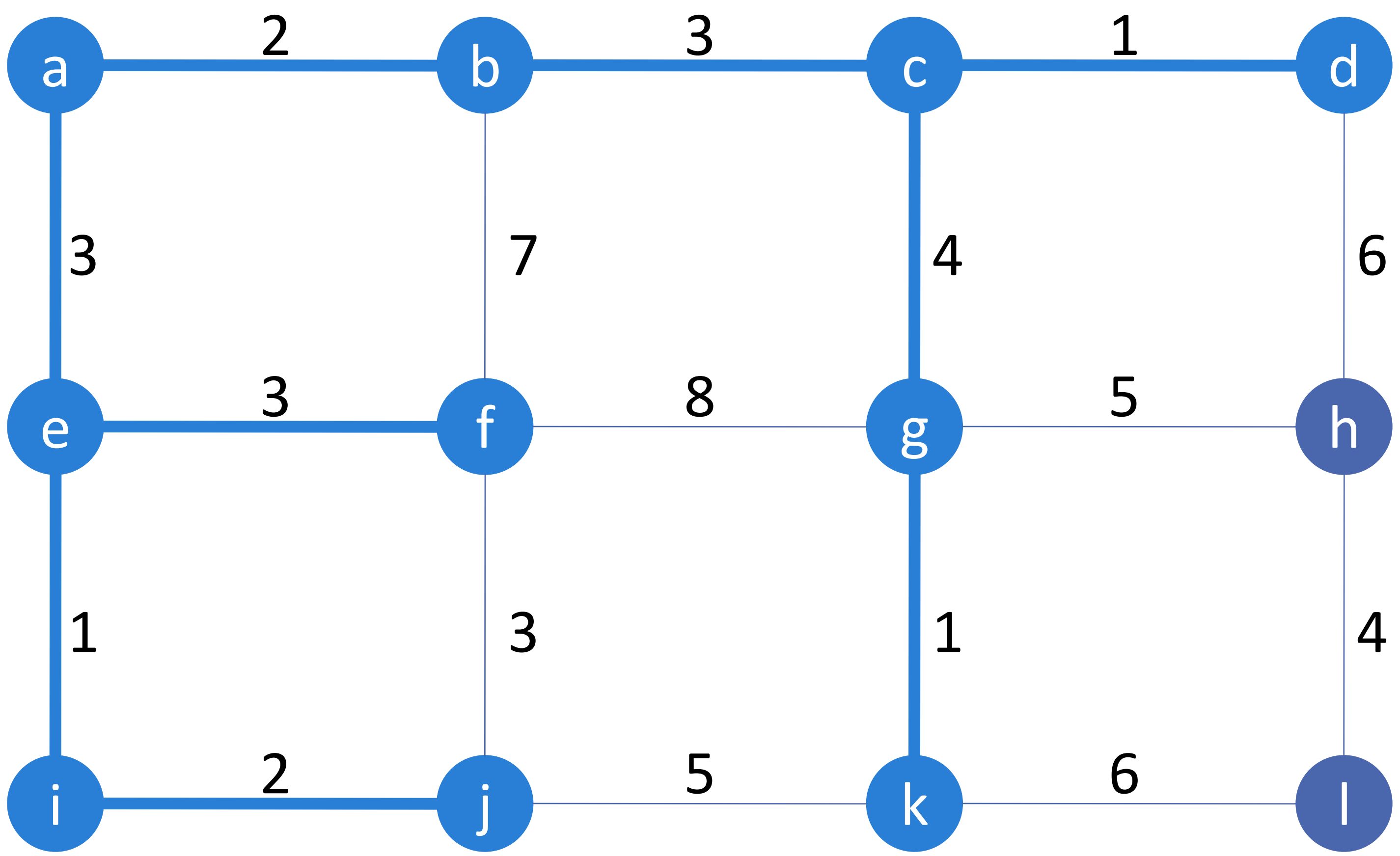
Ví dụ khác về giải thuật Prim



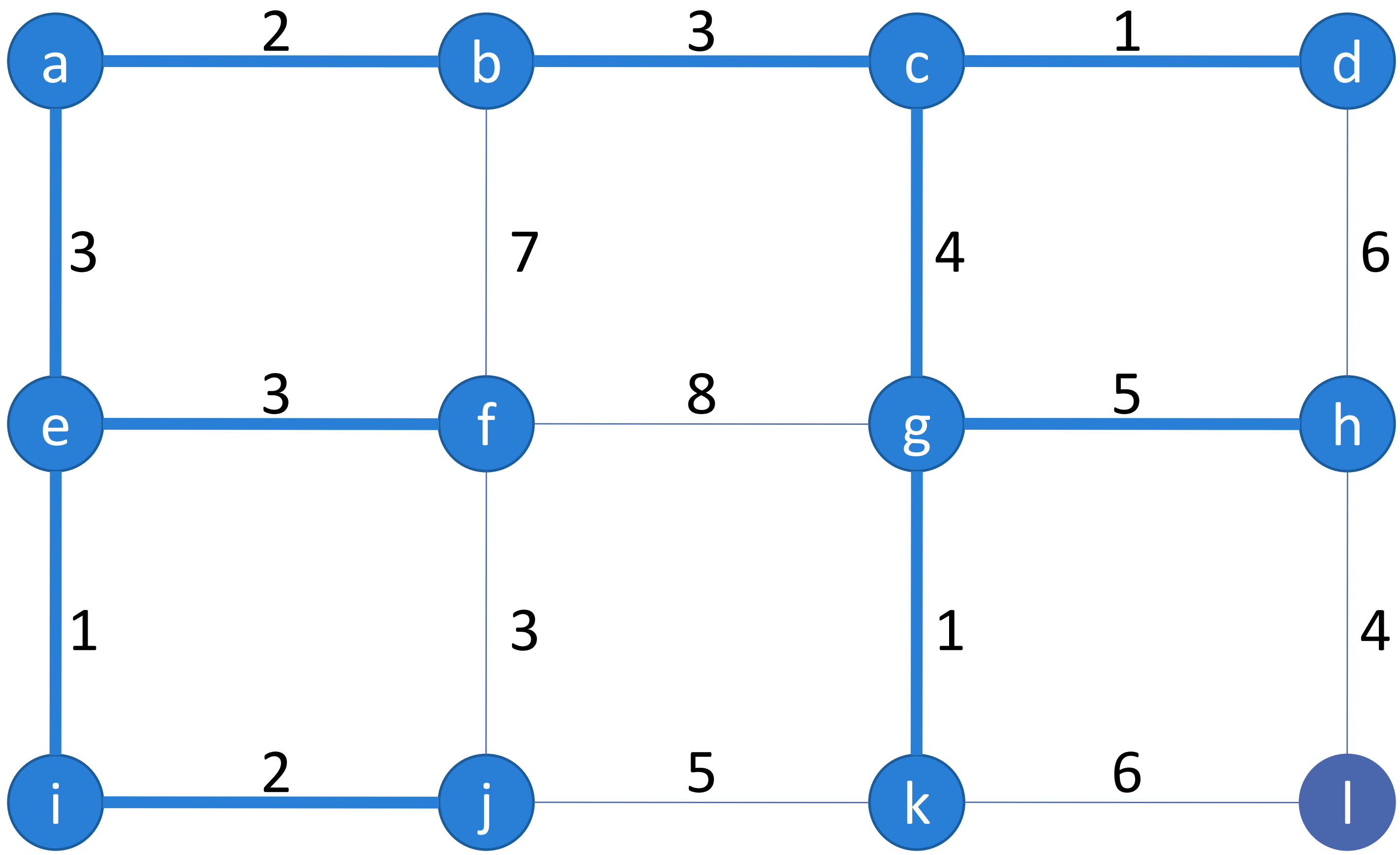
Ví dụ khác về giải thuật Prim



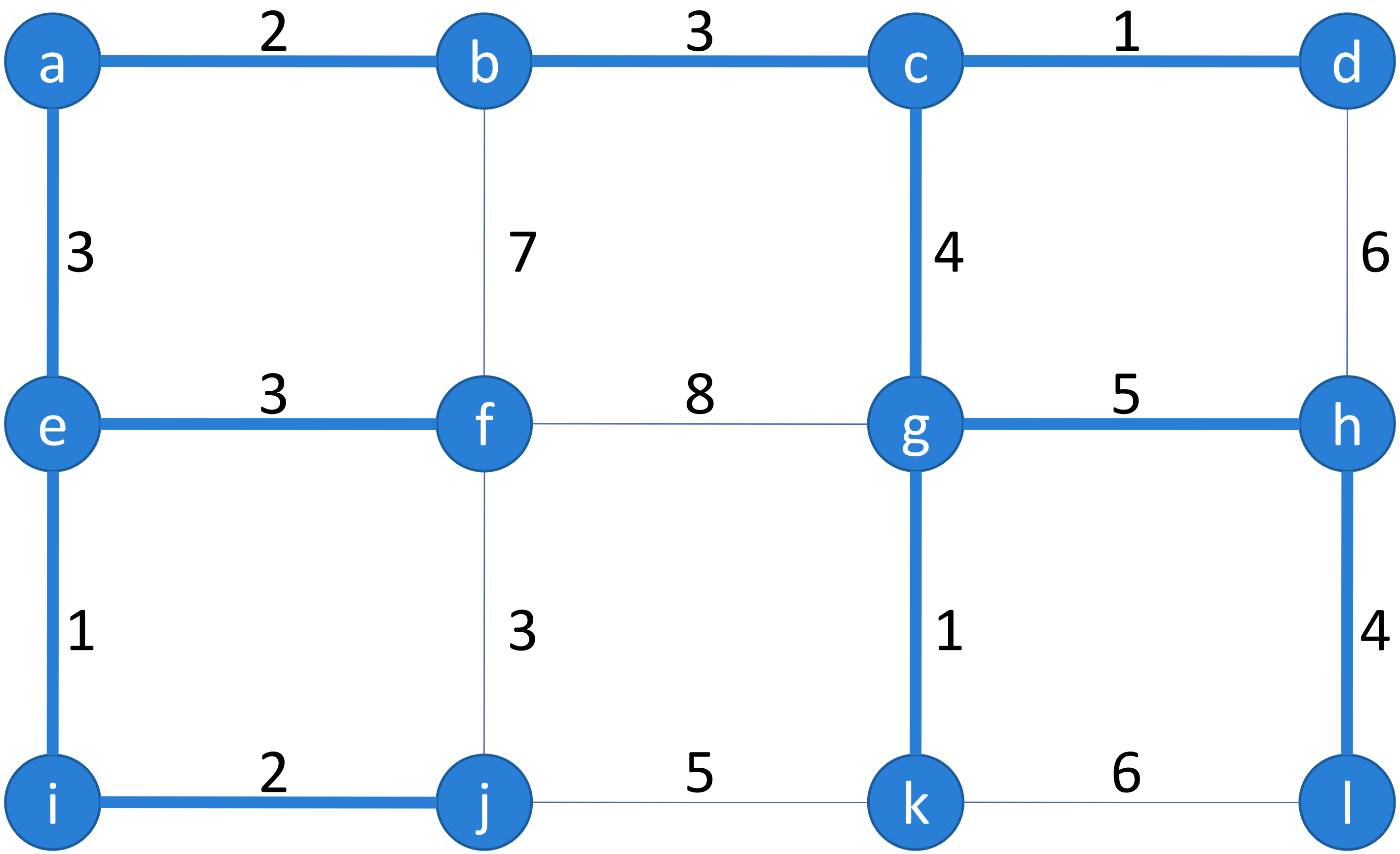
Ví dụ khác về giải thuật Prim



Ví dụ khác về giải thuật Prim

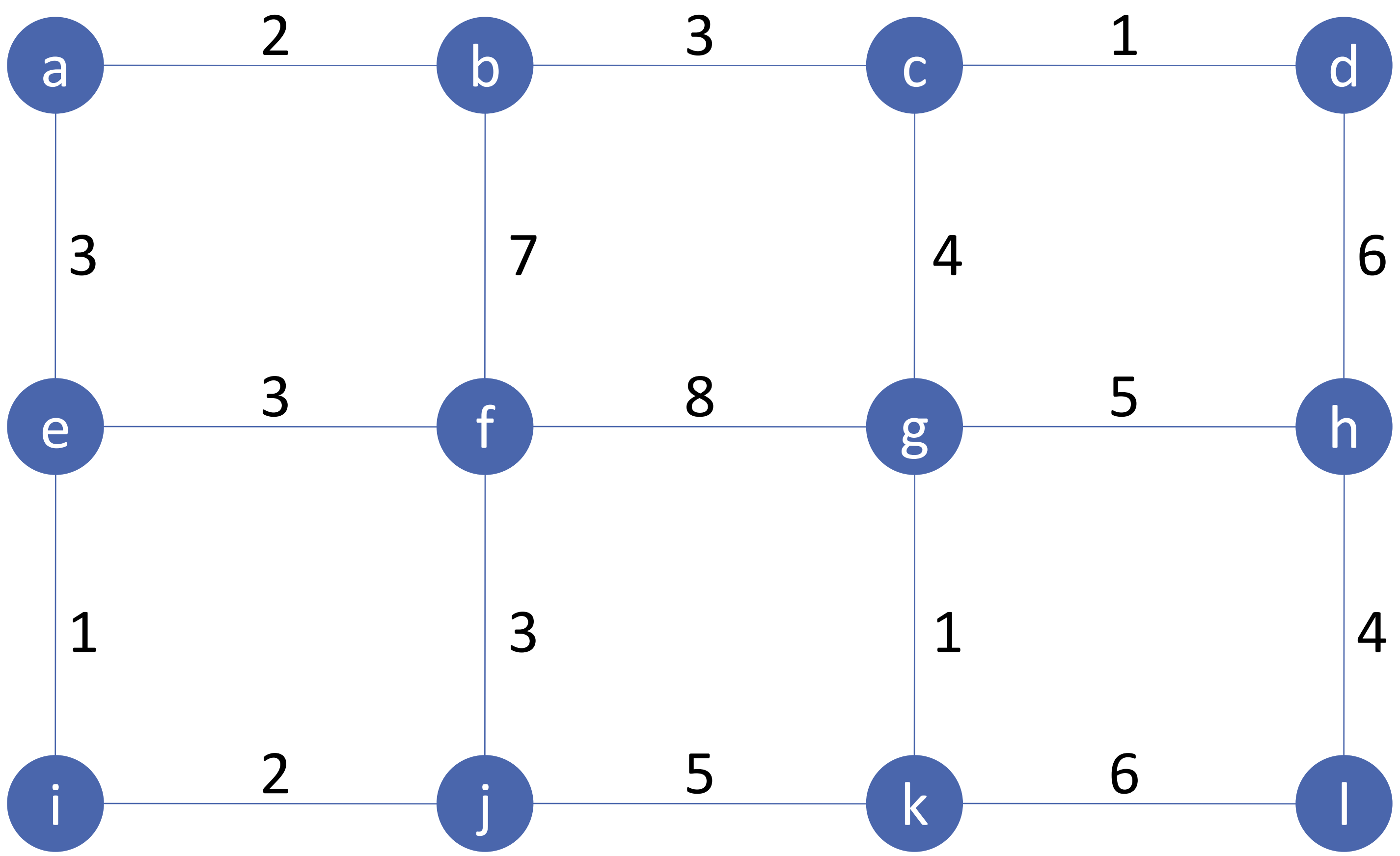


Ví dụ khác về giải thuật Prim

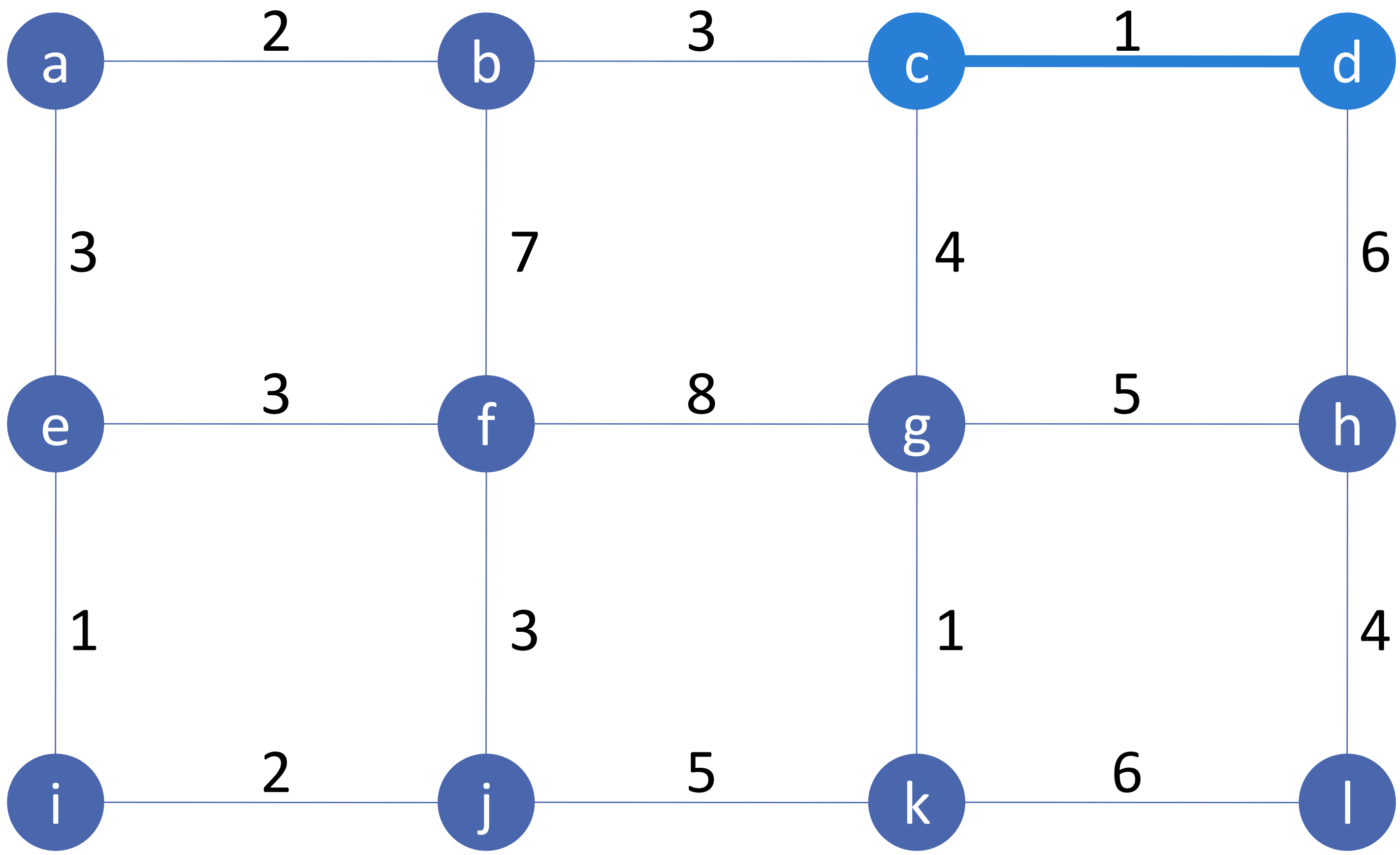


$$c(T) = 29$$

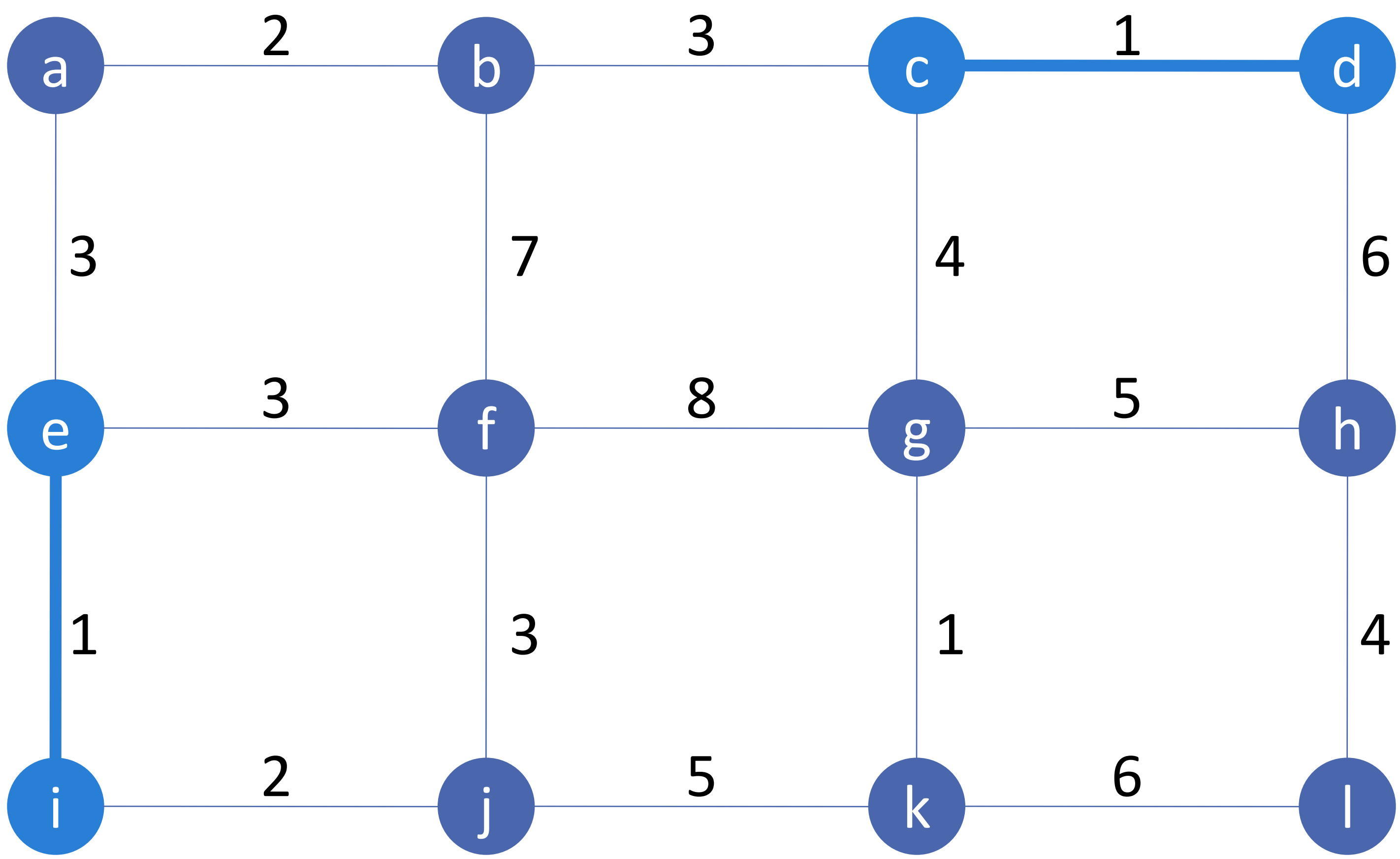
Ví dụ khác về giải thuật Kruskal



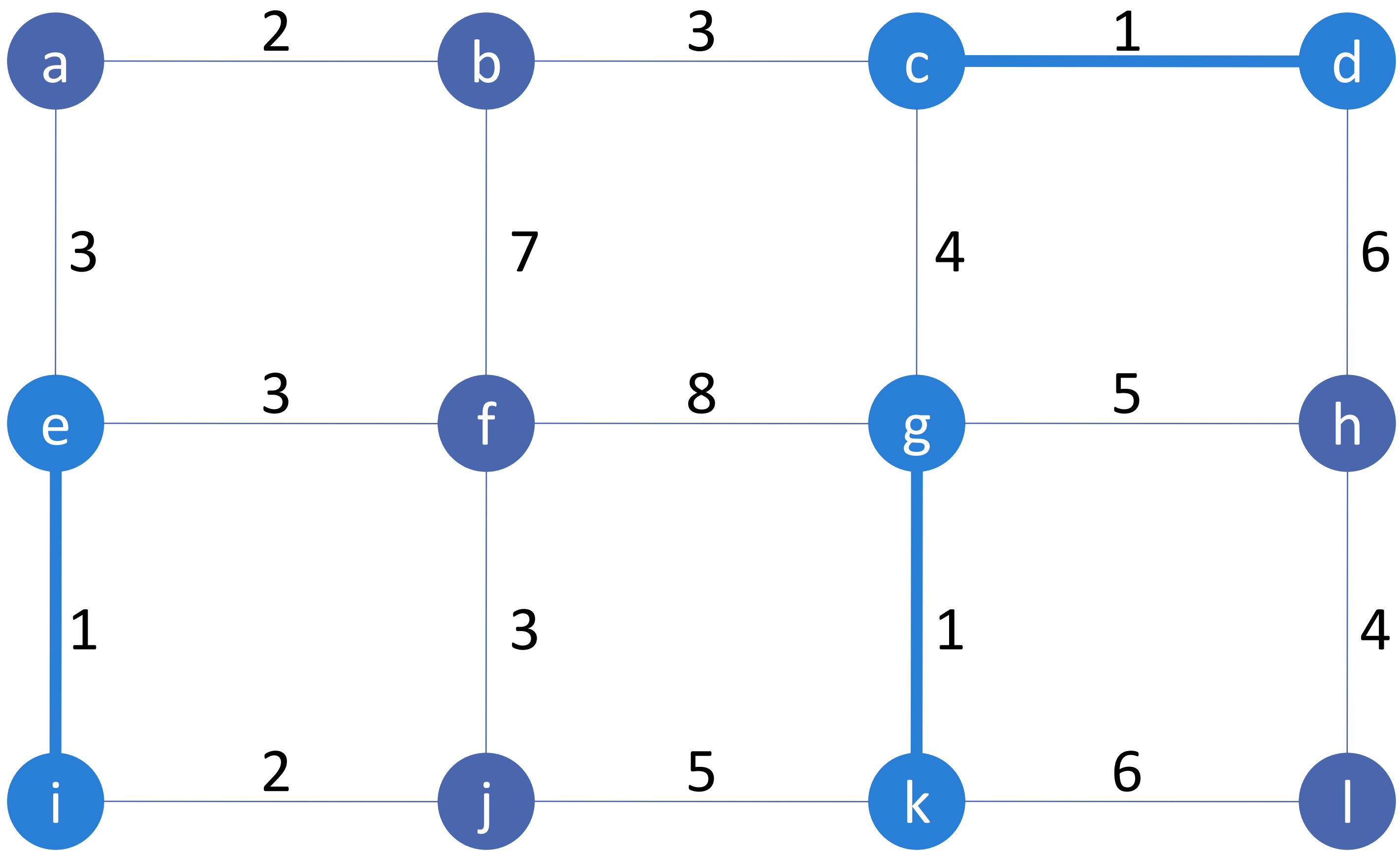
Ví dụ khác về giải thuật Kruskal



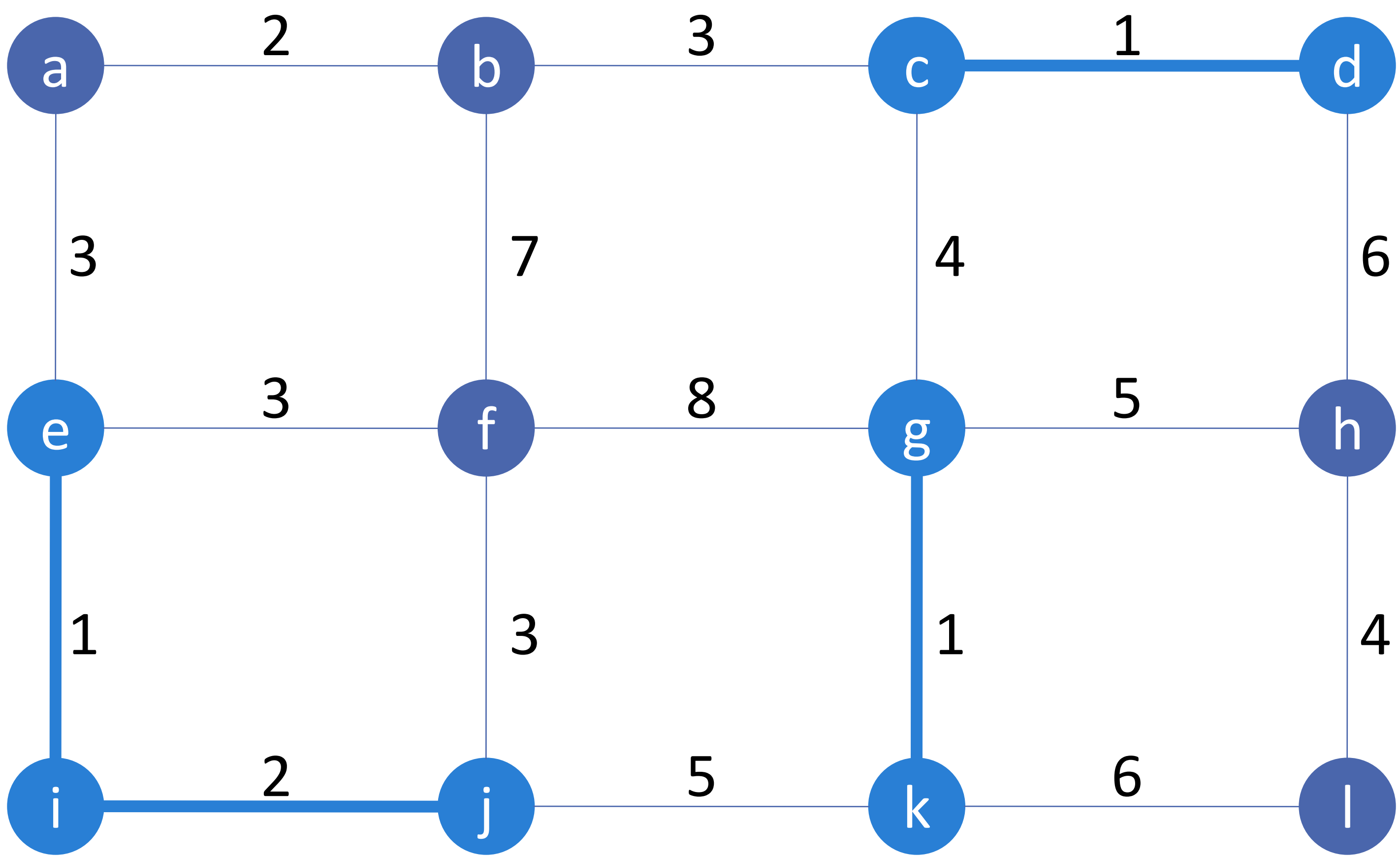
Ví dụ khác về giải thuật Kruskal



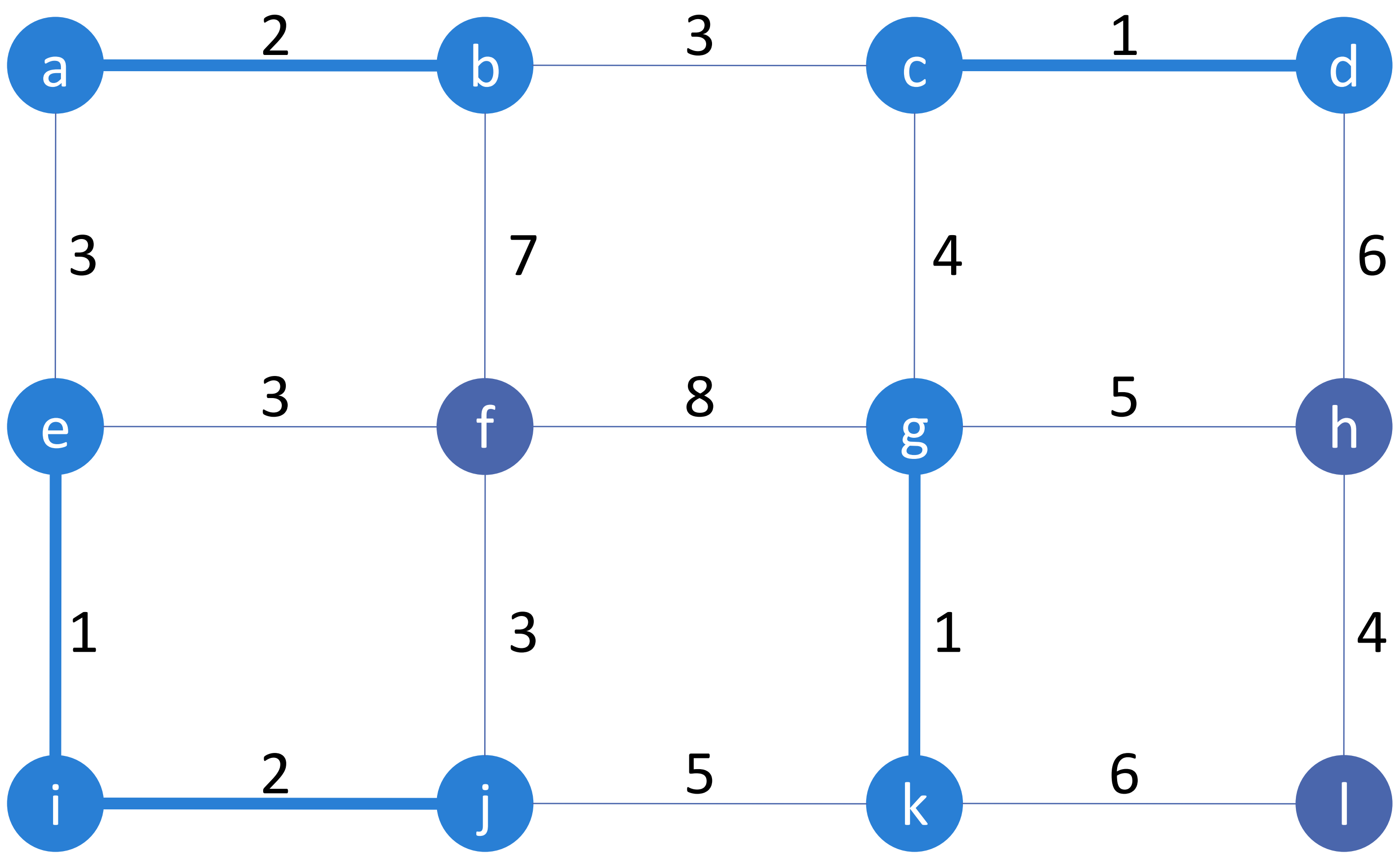
Ví dụ khác về giải thuật Kruskal



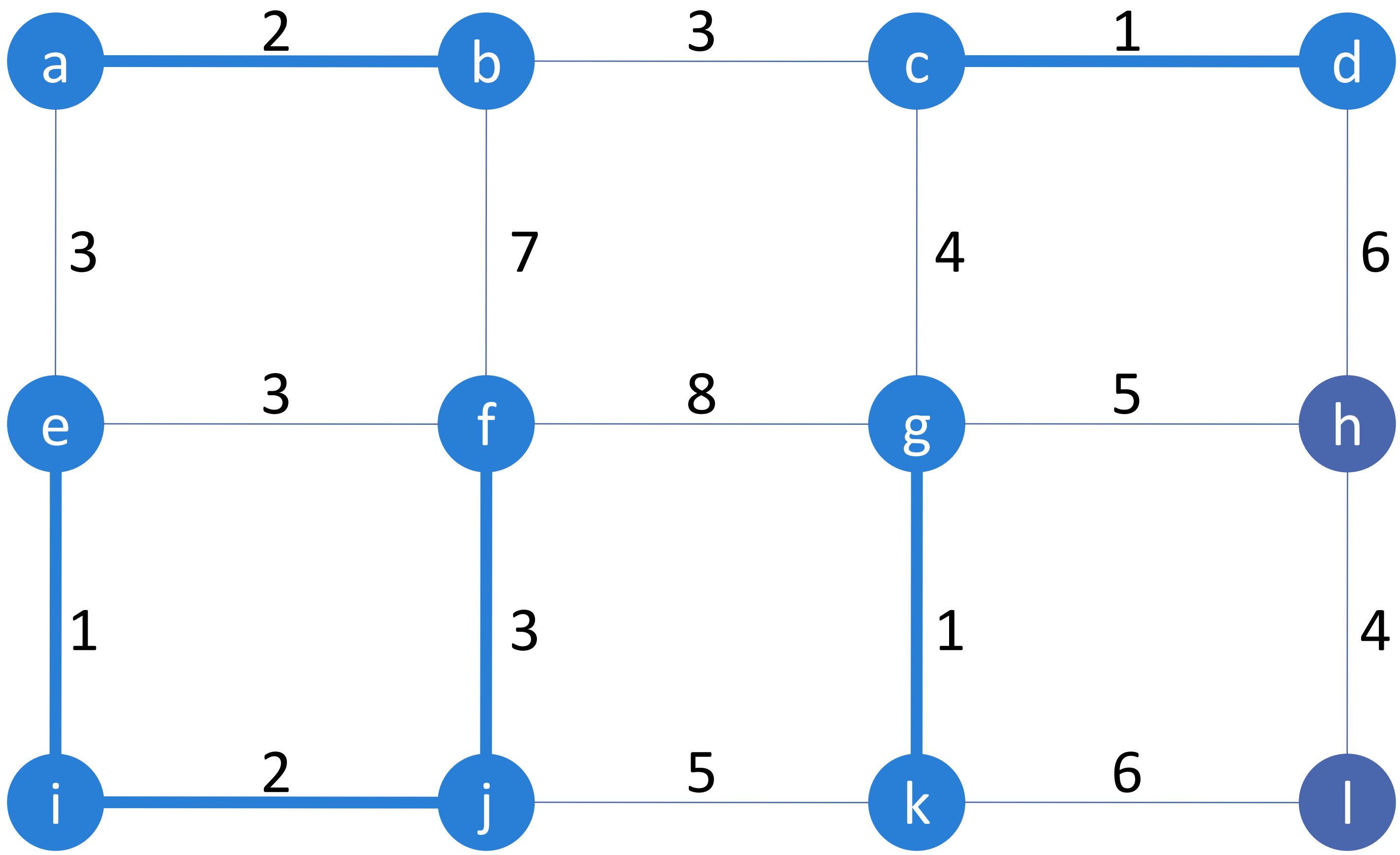
Ví dụ khác về giải thuật Kruskal



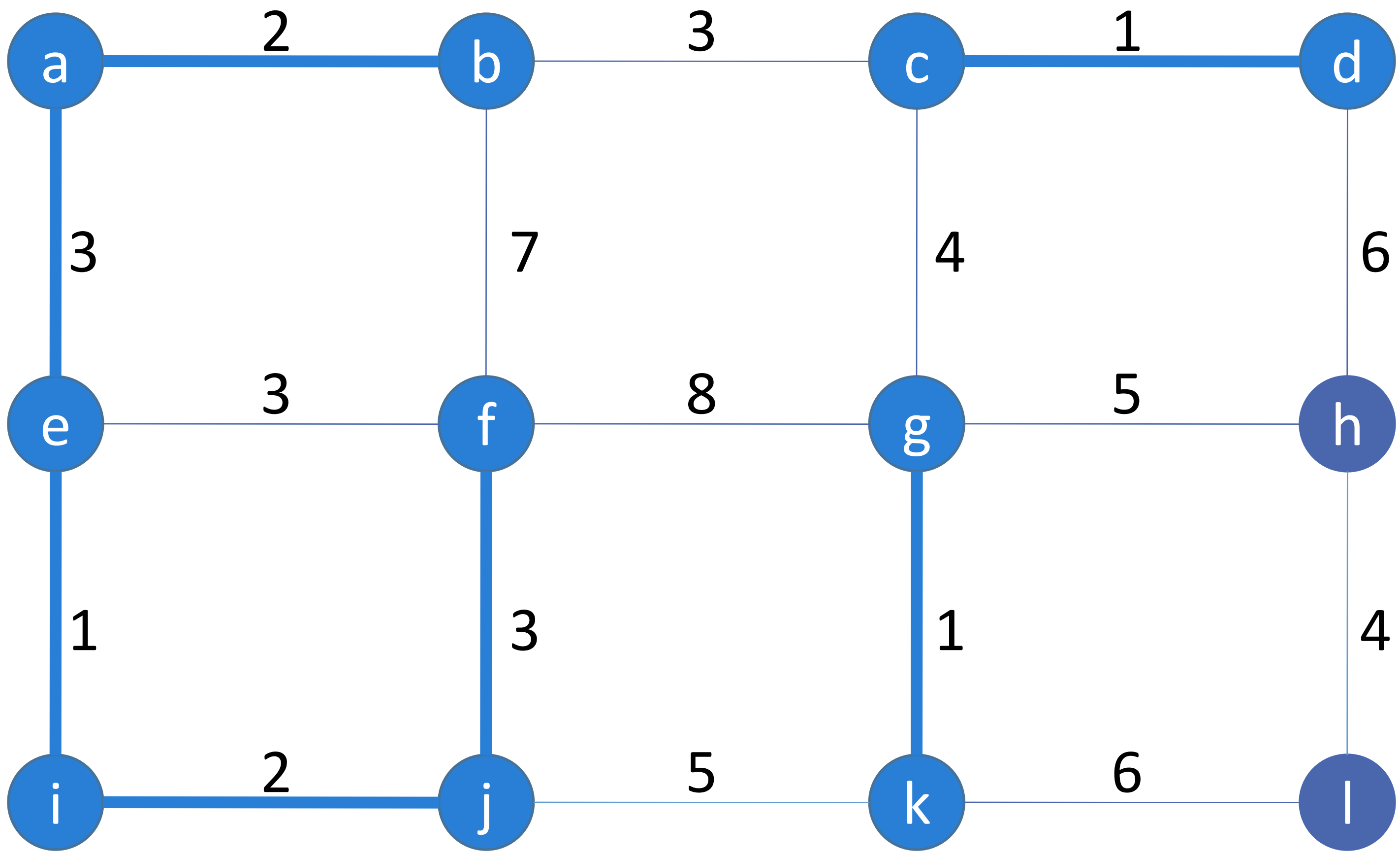
Ví dụ khác về giải thuật Kruskal



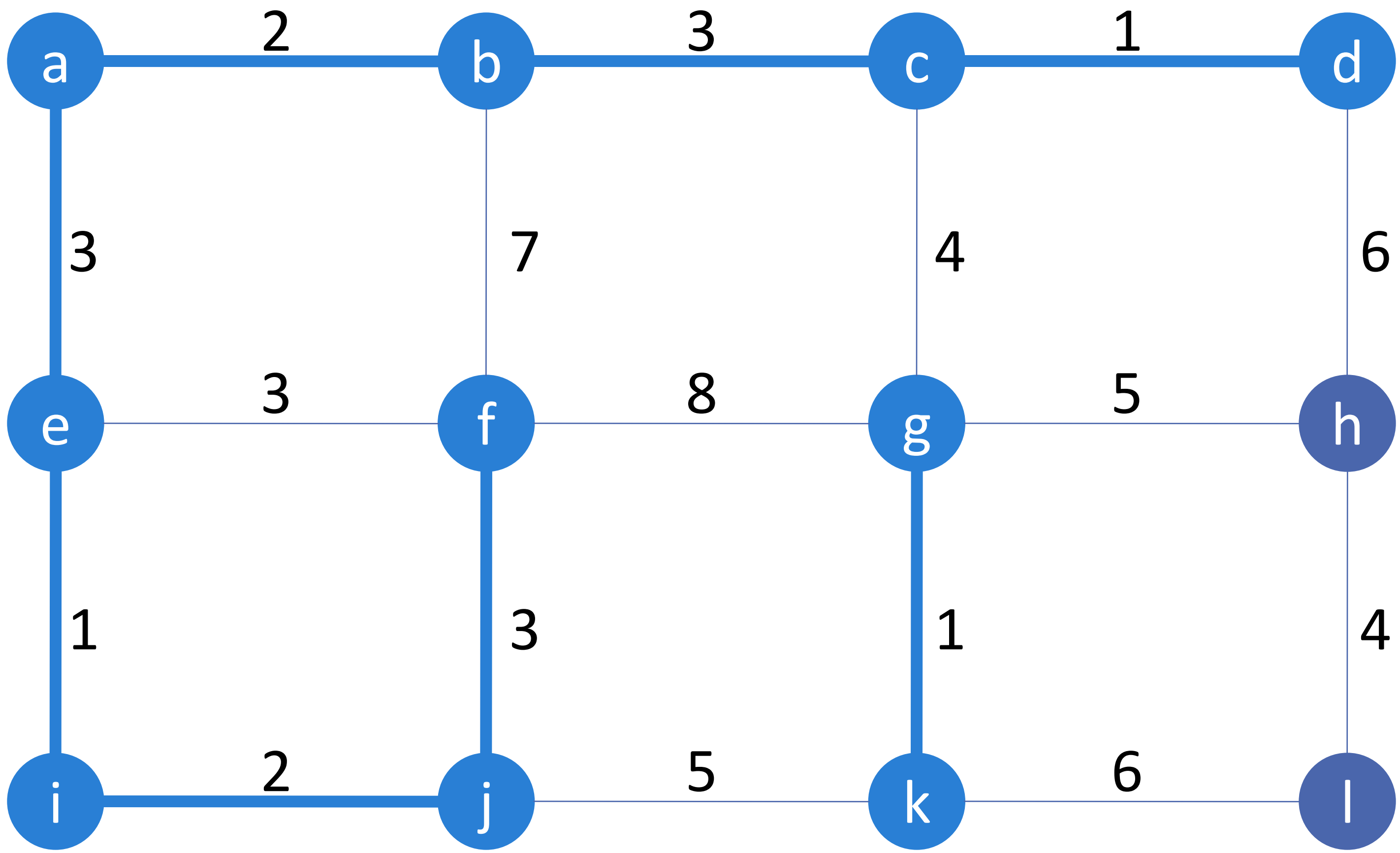
Ví dụ khác về giải thuật Kruskal



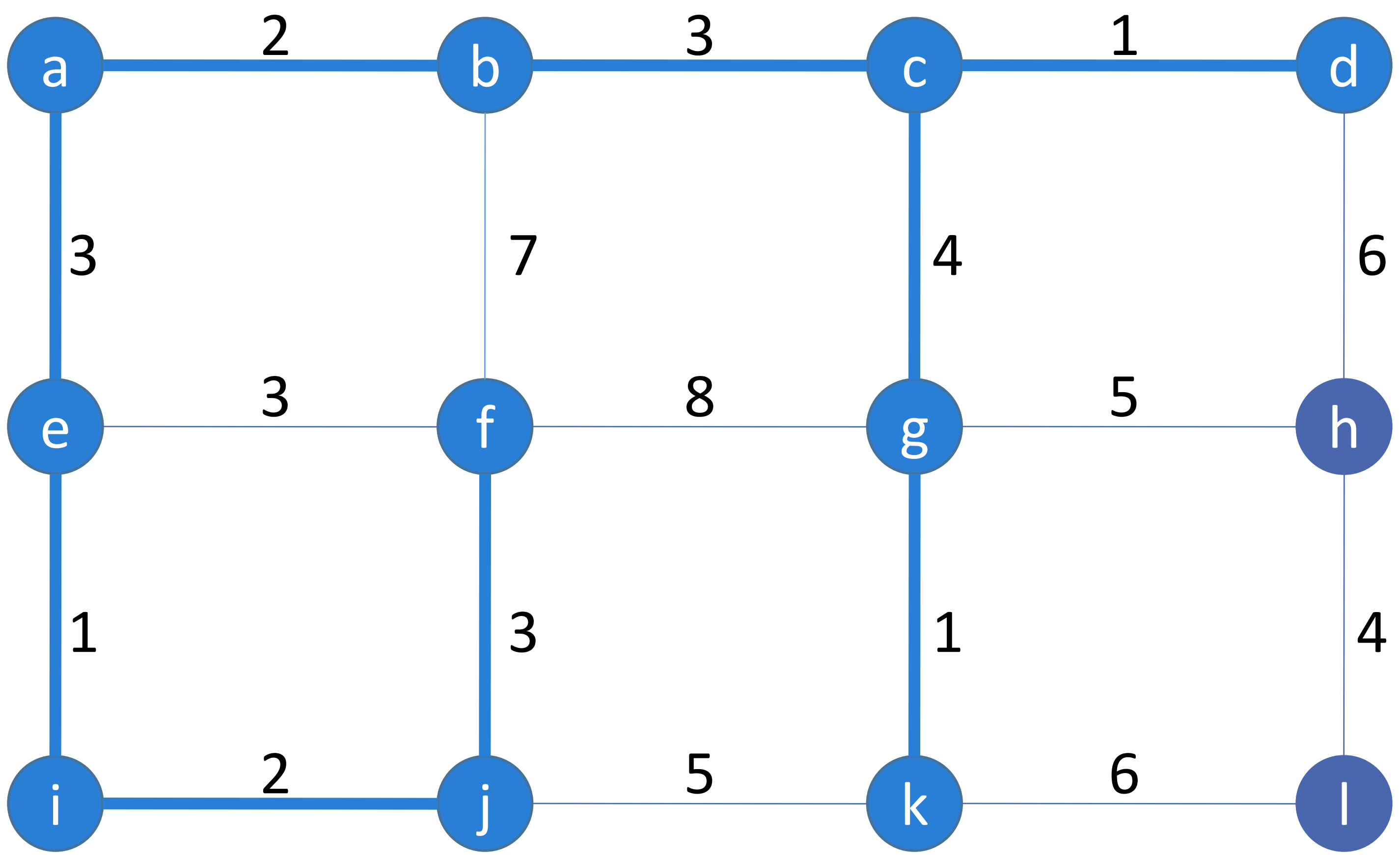
Ví dụ khác về giải thuật Kruskal



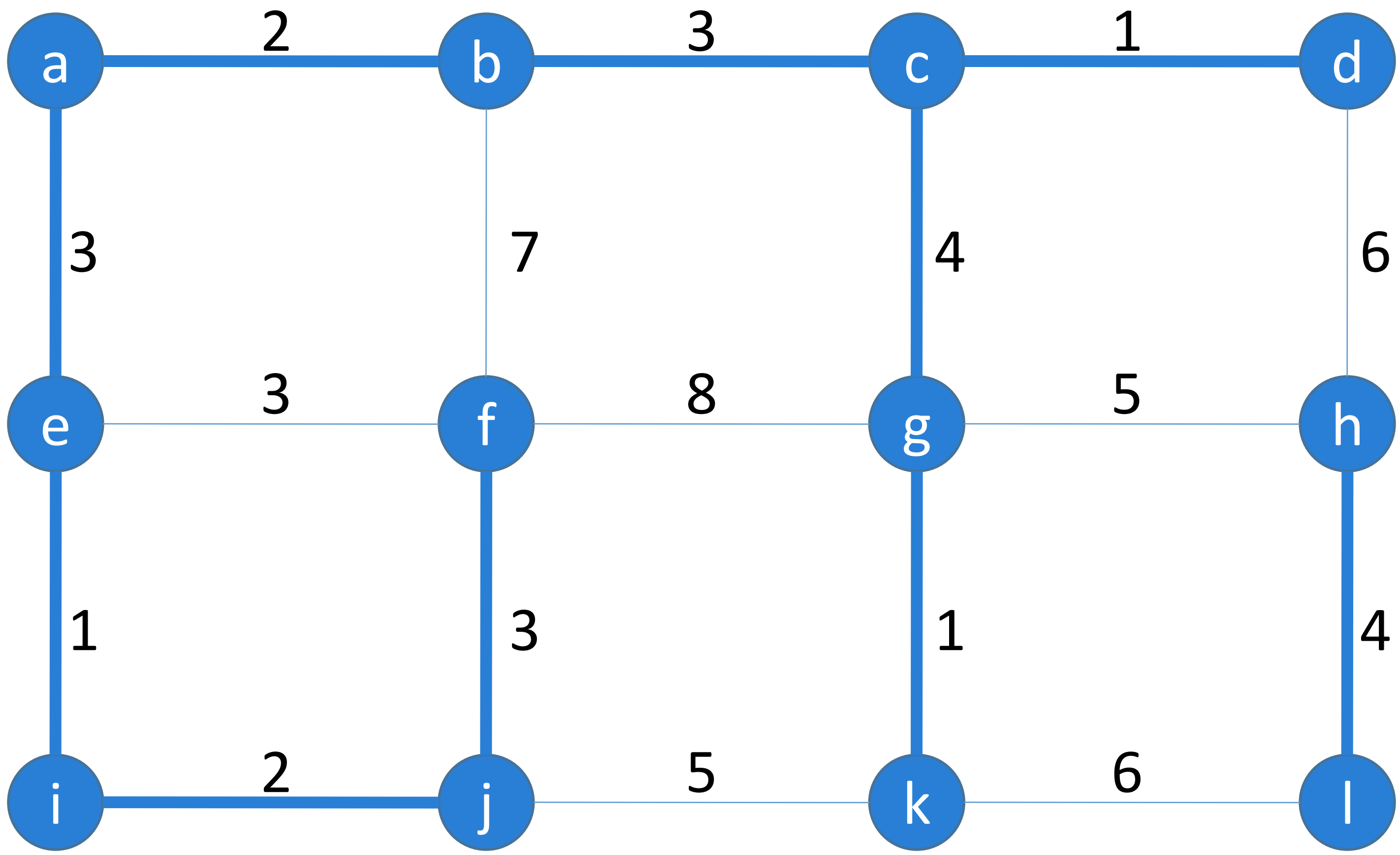
Ví dụ khác về giải thuật Kruskal



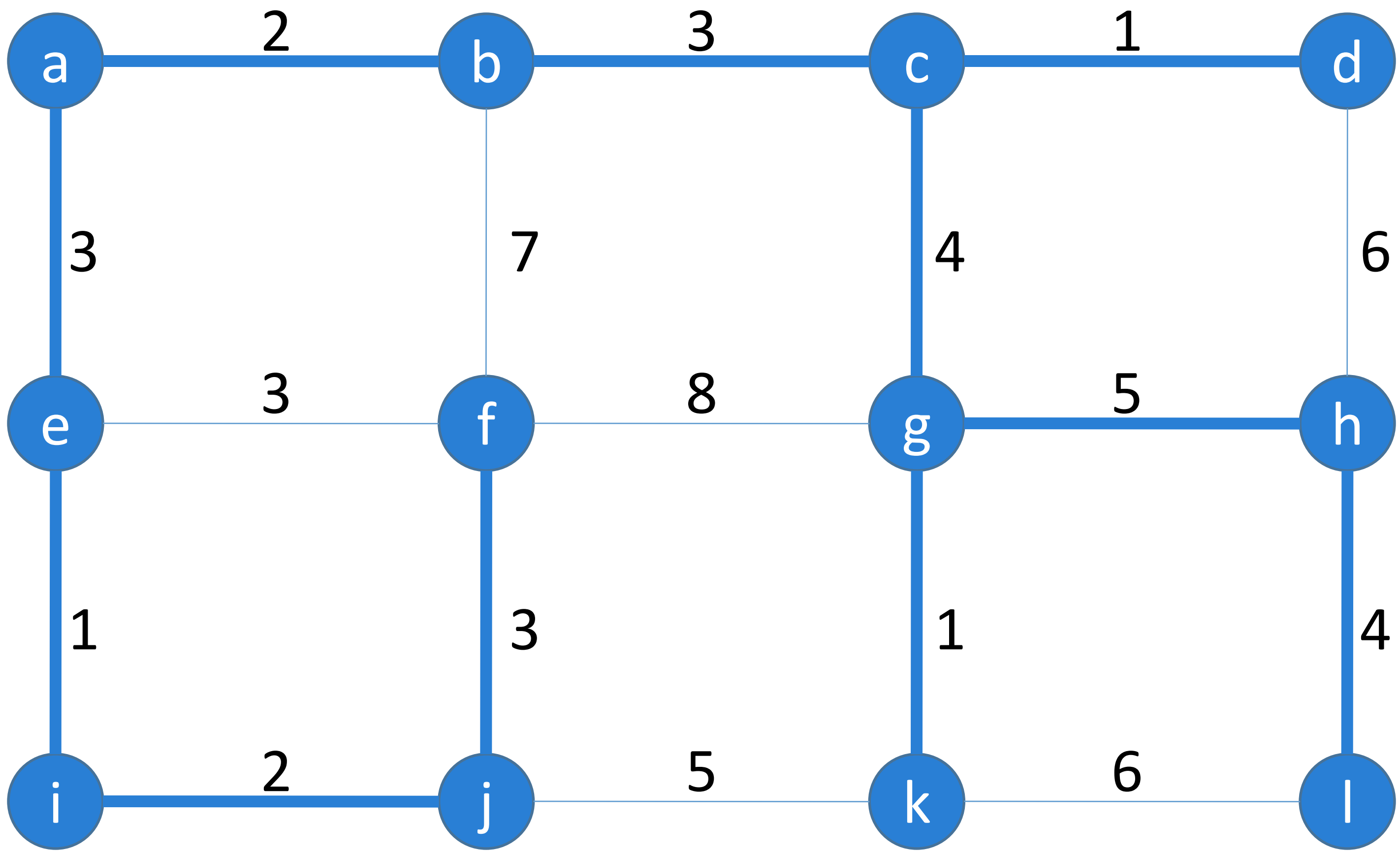
Ví dụ khác về giải thuật Kruskal



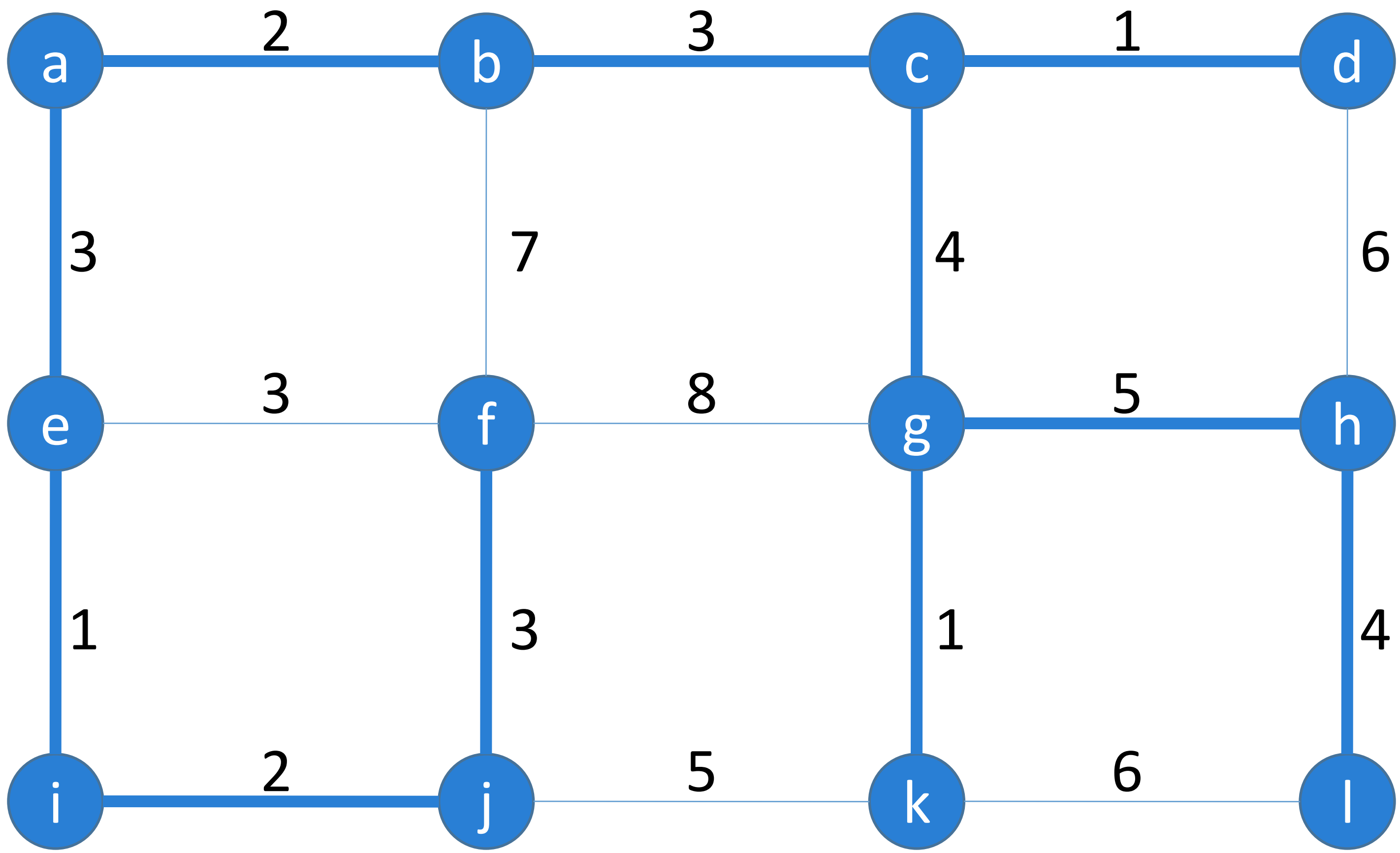
Ví dụ khác về giải thuật Kruskal



Ví dụ khác về giải thuật Kruskal

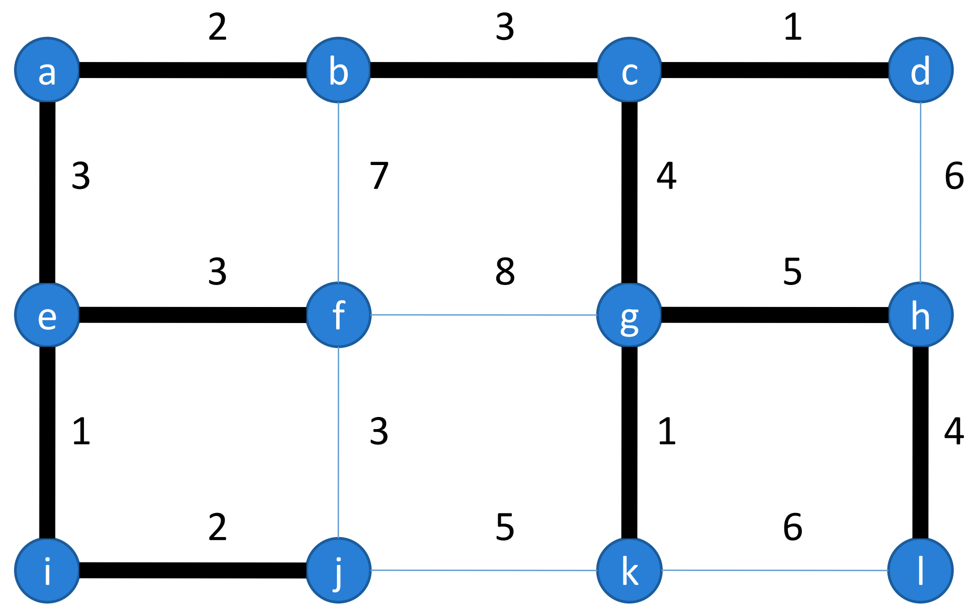


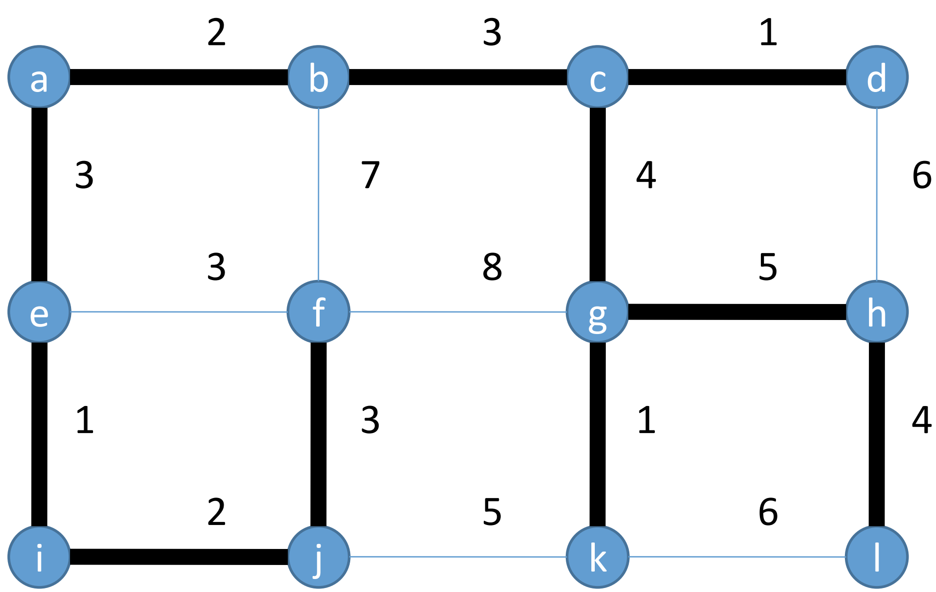
Ví dụ khác về giải thuật Kruskal



$$c(T) = 29$$

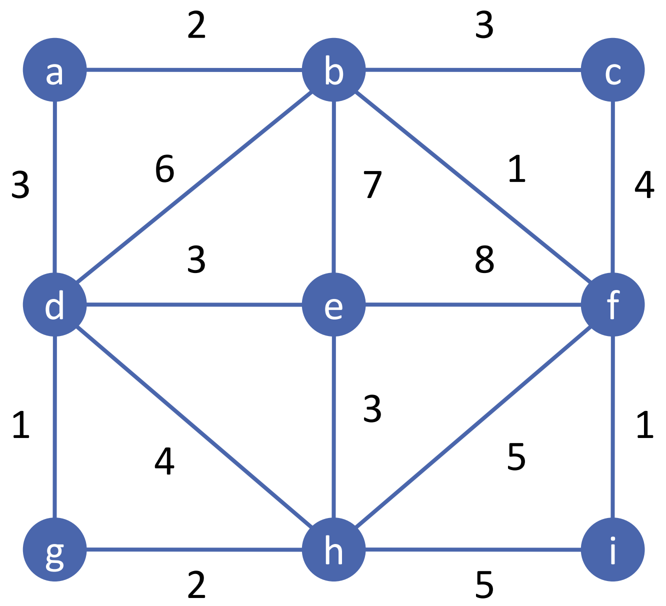
So sánh kết quả của Prim và Kruskal

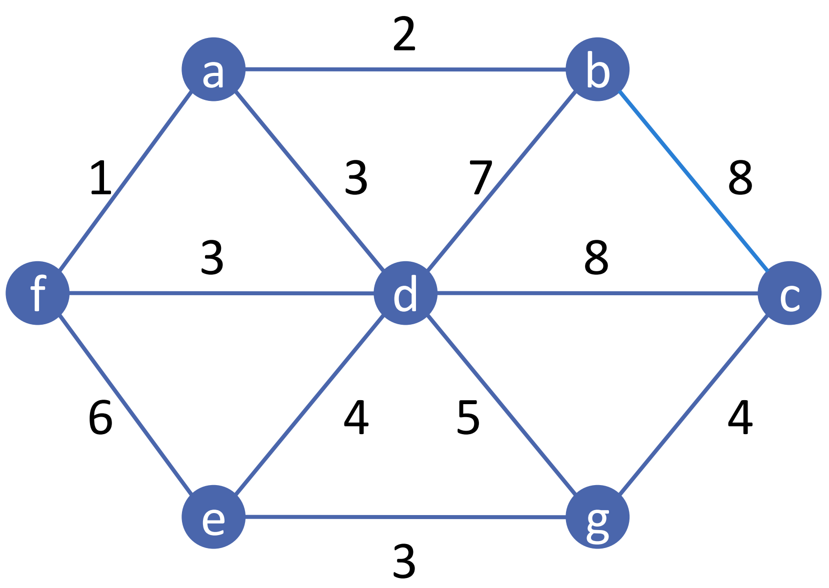




Bài tập rèn luyện

Sử dụng giải thuật Prim tìm cây khung nhỏ nhất cho các đồ thị có trọng số dưới đây. Thực hiện tương tự bằng giải thuật Kruskal.





Bài tập rèn luyện

Dưới đây là một số giải thuật tìm cây khung của đồ thị liên thông $\boldsymbol{G}$ cho trước. Hãy cho biết các giải thuật này có đúng không?

a) Bắt đầu với $\boldsymbol{G}$. Nếu $\boldsymbol{G}$ có chu trình, hủy một cạnh thuộc chu trình này khỏi $\boldsymbol{G}$. Lặp lại thủ tục này đến khi đồ thị nhận được không còn chu trình.

b) Bắt đầu bằng một cạnh, thêm dần mỗi lần một cạnh mới cho đến khi đồ thị nhận được có $n-1$ cạnh.

c) Bắt đầu bằng một cạnh, thêm dần mỗi lần một cạnh mới với điều kiện không tạo ra chu trình cho đến khi không thể thêm được cạnh mới nào nữa.


...the end.