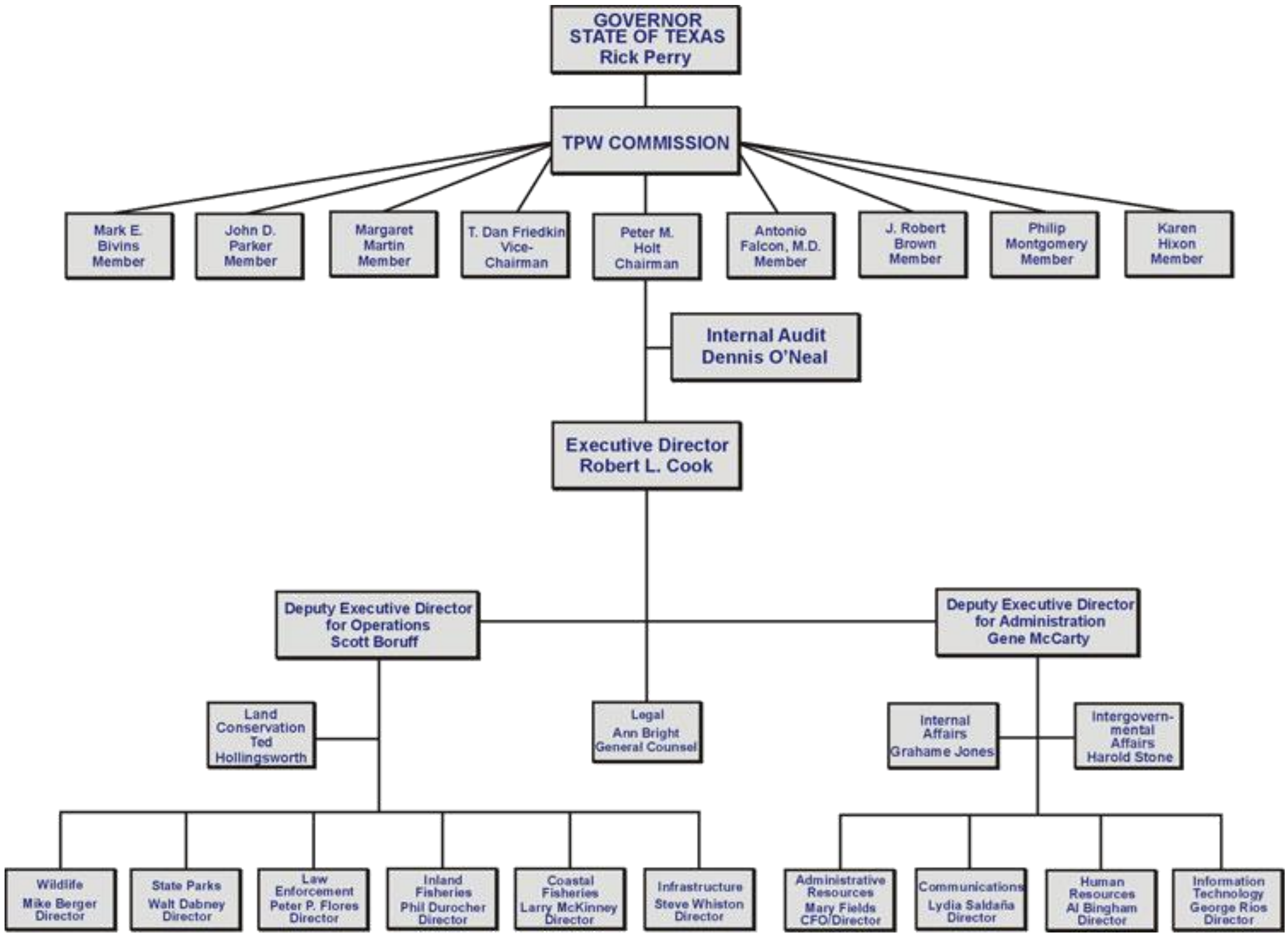
03. ĐỒ THỊ DẠNG CÂY

Bài giảng Lý thuyết đồ thị

Định nghĩa

- Cây (tree) là một đồ thị vô hướng, liên thông và không có chu trình.
- Định nghĩa cây hàm ý rằng, mọi cây đều không chứa khuyên cũng như cạnh song song (vì khuyên và cạnh song song sẽ tạo thành chu trình).

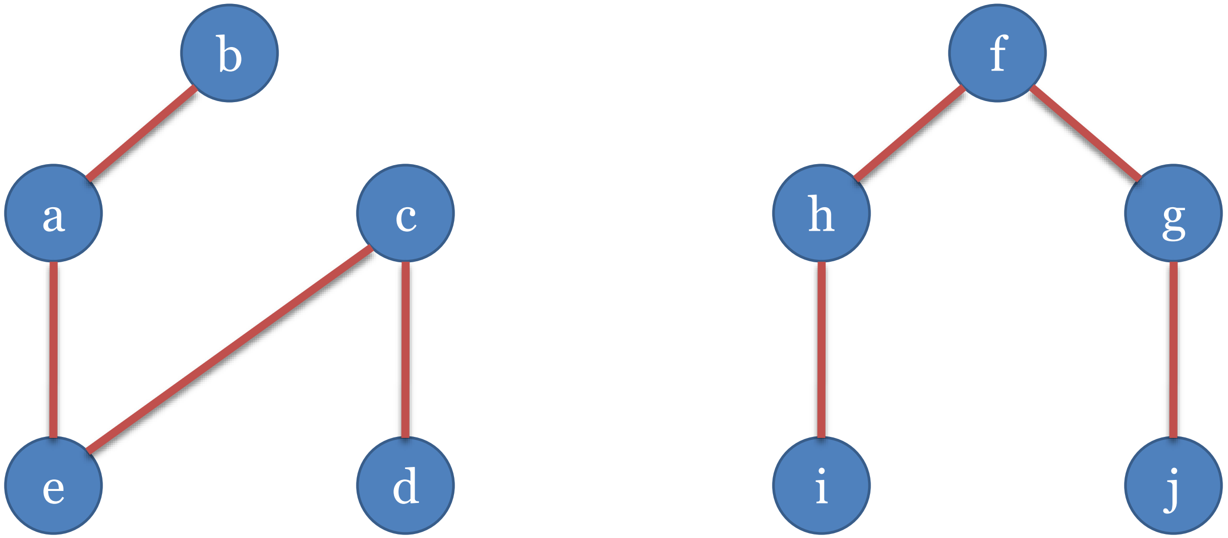
Ví dụ về cây



GOVERNOR
STATE OF TEXAS
Rick Perry
TPW COMMISSION
Mark E. Bivins Member
John D. Parker Member
Margaret Martin Member
T. Dan Friedkin Vice-Chairman
Peter M. Holt Chairman
Antonio Falcon, M.D. Member
J. Robert Brown Member
Philip Montgomery Member
Karen Hixon Member
Internal Audit
Dennis O'Neal
Executive Director
Robert L. Cook
Deputy Executive Director for Operations Scott Boruff
Deputy Executive Director for Administration Gene McCarty
Land Conservation Ted Hollingsworth
Legal Ann Bright General Counsel
Internal Affairs Grahame Jones
Intergovern-mental Affairs Harold Stone
Wildlife Mike Berger Director
State Parks Walt Dabney Director
Law Enforcement Peter P. Flores Director
Inland Fisheries Phil Durocher Director
Coastal Fisheries Larry McKinney Director
Infrastructure Steve Whiston Director
Administrative Resources Mary Fields CFO/Director
Communications Lydia Saldaña Director
Human Resources Al Bingham Director
Information Technology George Rios Director

Rừng

- Rừng (forest) là một đồ thị mà mỗi thành phần liên thông của nó là một cây.
- Suy ra, đồ thị không có chu trình là một rừng



CÁC TÍNH CHẤT CỦA CÂY

Định lý về sự tồn tại các đỉnh treo

- Nếu một cây T gồm n đỉnh với $n \geq 2$ thì T chứa ít nhất 2 đỉnh treo.

Chứng minh:

- Gọi v_1, v_2… v_k là đường đi đơn dài nhất trên cây. Khi đó rõ ràng v_1 và v_k là các đỉnh treo vì từ v_1(hoặc v_k) sẽ không có cạnh nối tới bất kỳ đỉnh nào trong số các đỉnh v_2, v_3.. 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 T phải chứa ít nhất 2 đỉnh treo.

Các tính chất của cây

- Đơn đồ thị T = (V, E) là một đồ thị vô hướng n đỉnh. Khi đó, các mệnh đều sau đây tương đương:
  1. T là một 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 1 ➔ 2

T là cây ➔ T không chứa chu trình và có n – 1 cạnh.

- Với n = 1, tính chất đúng vì khi đó cây T có 0 cạnh.
- Giả sử n = k đúng, nghĩa là có k – 1 cạnh.
- Ta chứng minh cây T có n = k+1 đỉnh sẽ có k cạnh.
  - Thật vậy, do định lý về sự tồn tại các đỉnh treo, nên tồn tại ít nhất 2 đỉnh treo trong T. Giả sử 2 đỉnh treo đó là v_1 và v_2. Loại bỏ đỉnh v_1 và cạnh nối với nó. Khi đó, ta có đồ thị còn lại là một cây có k đỉnh, do đó theo giả thiết quy nạp, nó có k – 1 cạnh.
  - Vậy cây ban đầu có k cạnh. [đpcm]

Chứng minh 2 ➔ 3

- *T không chứa chu trình và có n – 1 cạnh ➔ T liên thông (và có n – 1 cạnh).*

- Giả sử T không liên thông: T có k ≥ 2 thành phần liên thông T_1 T_2… T_k. Do T không có chu trình nên các thành phần liên thông T_i cũng không có chu trình. Do 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(T1) + … + e(Tk)
  = n(T1) + … + n(Tk) – k
  = n(T) – k = n – k < n – 1
  (mâu thuẫn vì e(T) = n - 1).
- Do vậy, T liên thông (đpcm).

Chứng minh 3 → 4

T liên thông và có n – 1 cạnh → Mỗi cạnh của T đều là cầu (và T liên thông)

- Khi ta loại bỏ một cạnh bất kỳ khỏi T, lúc đó số cạnh của T là n – 2, nên với T có n đỉnh, n – 2 cạnh chắc chắn sẽ không liên thông. Nói cách khác, 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 4 → 5

T liên thông và mỗi cạnh đều là cầu → Hai đỉnh bất kỳ của T được nối với nhau bởi đúng 1 đường đi.

- Do T liên thông, nên theo định nghĩa 2 đỉnh bất kỳ của nó đều có thể nối với nhau bởi một đường đi đơn.
- Nếu có cặp đỉnh nào của T có 2 đường đi đơn khác nhau nối chúng thì từ đó suy ra đồ thị chứa chu trình, vì thế 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 5 ➔ 6

Hai đỉnh bất kỳ của T được nối với nhau bởi đúng 1 đường đi đơn ➔ T không chứa chu trình, nhưng khi thêm vào một cạnh ta thu được đúng một chu trình.

1. T không chứa chu trình, vì nếu có, ta sẽ tìm được 2 đỉnh có 2 đường đi đơn phân biệt nối với nhau.
2. Nếu ta thêm cạnh (u, v) bất kỳ, khi đó cạnh này và đường đi đơn từ u đến v hiện có sẽ tạo thành chu trình. Chu trình này là duy nhất vì nếu thu được nhiều hơn 1 chu trình thì chứng tỏ rằng trước đó trong T đã có chu trình.

Chứng minh 6 → 1

T không chứa chu trình, nhưng hễ cứ thêm một cạnh bất kỳ vào T thì ta thu được đúng một chu trình → T là cây (T liên thông và không có chu trình)

- Giả sử T không liên thông. Khi đó T có ít nhất 2 thành phần liên thông. Vì vậy, nếu thêm vào T một cạnh nối 2 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

1. Vẽ tất cả các cây không đẳng cấu có:

a) 4 đỉnh

b) 5 đỉnh

c) 6 đỉnh

Bài tập

2. 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

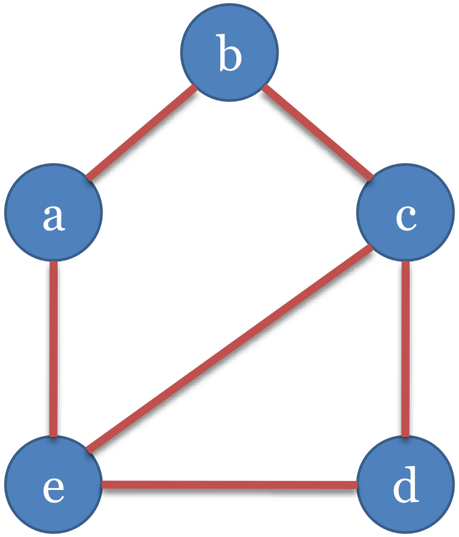
4. Đồ thị G là một rừng gồm t cây và có n đỉnh. Tìm số cạnh của G.

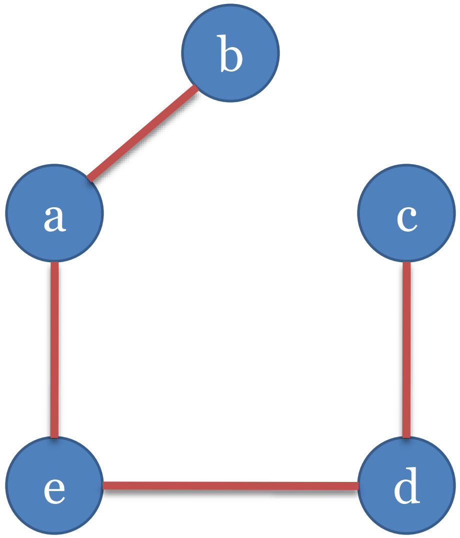
CÂY KHUNG CỦA ĐỒ THỊ

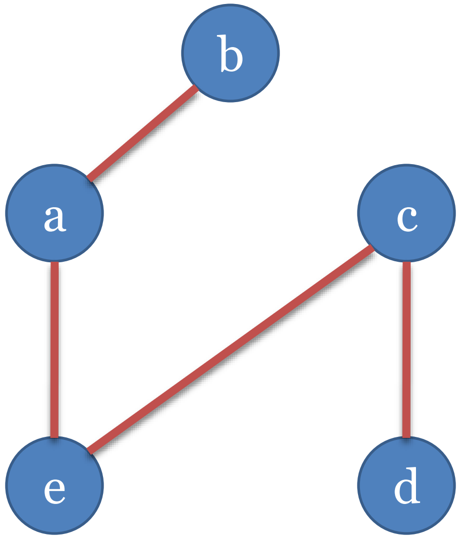
Định nghĩa

- 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.

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

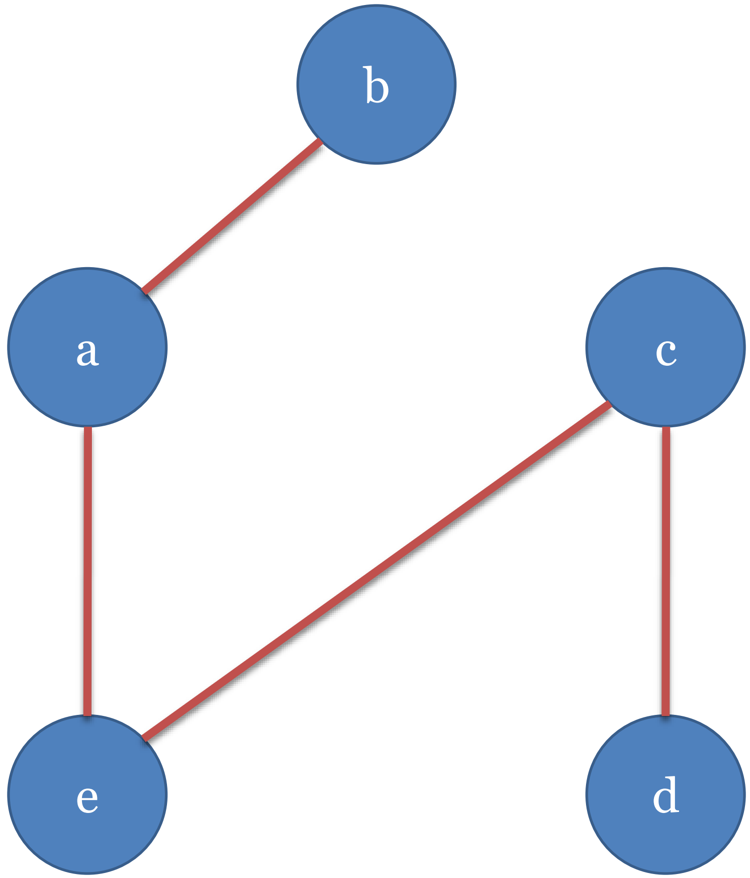
Định lý về tính liên thông

- 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?

Bài tập

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

Bài tập

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	

Bài tập

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

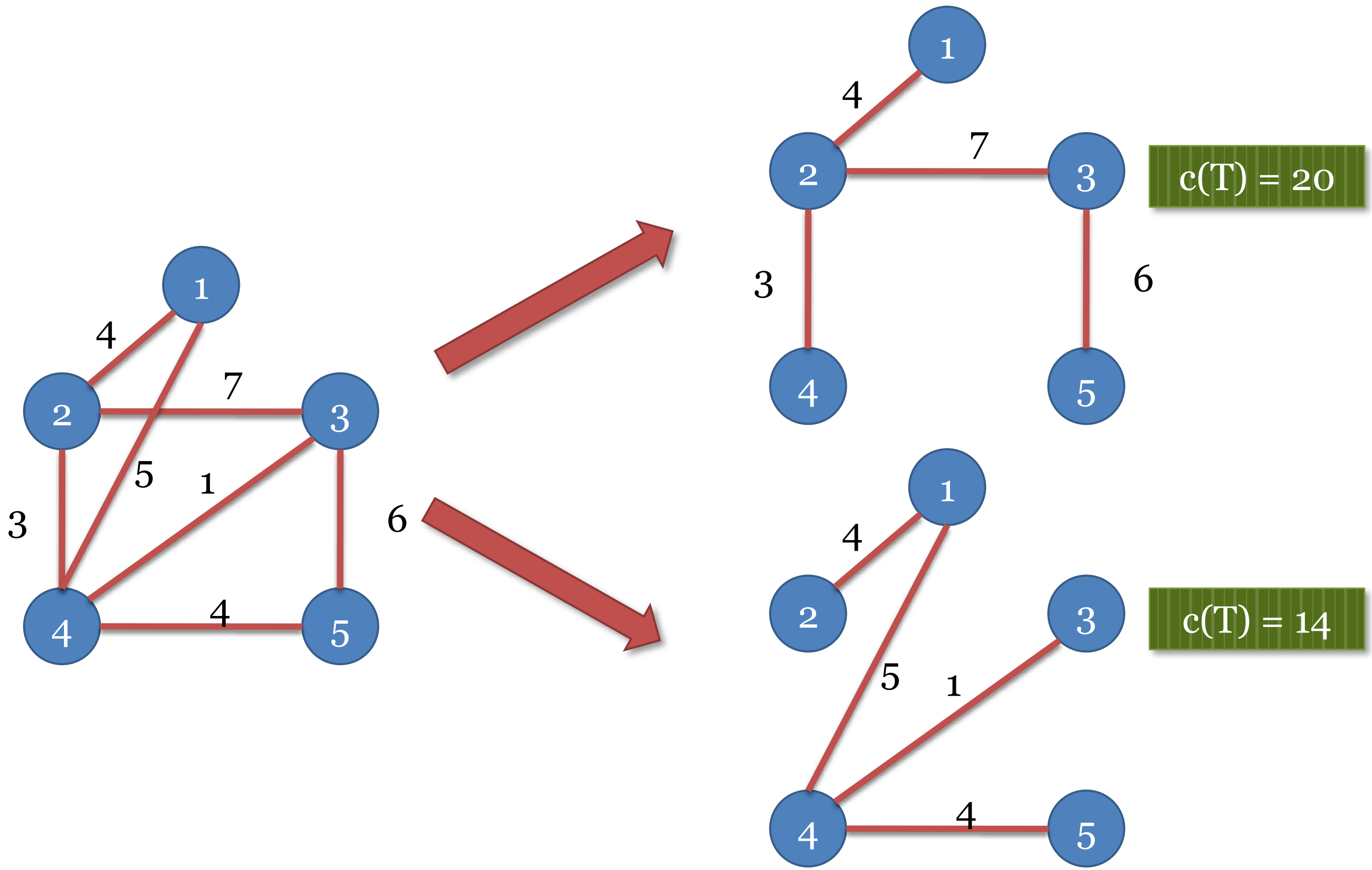
Độ dài của cây khung

- 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) 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).

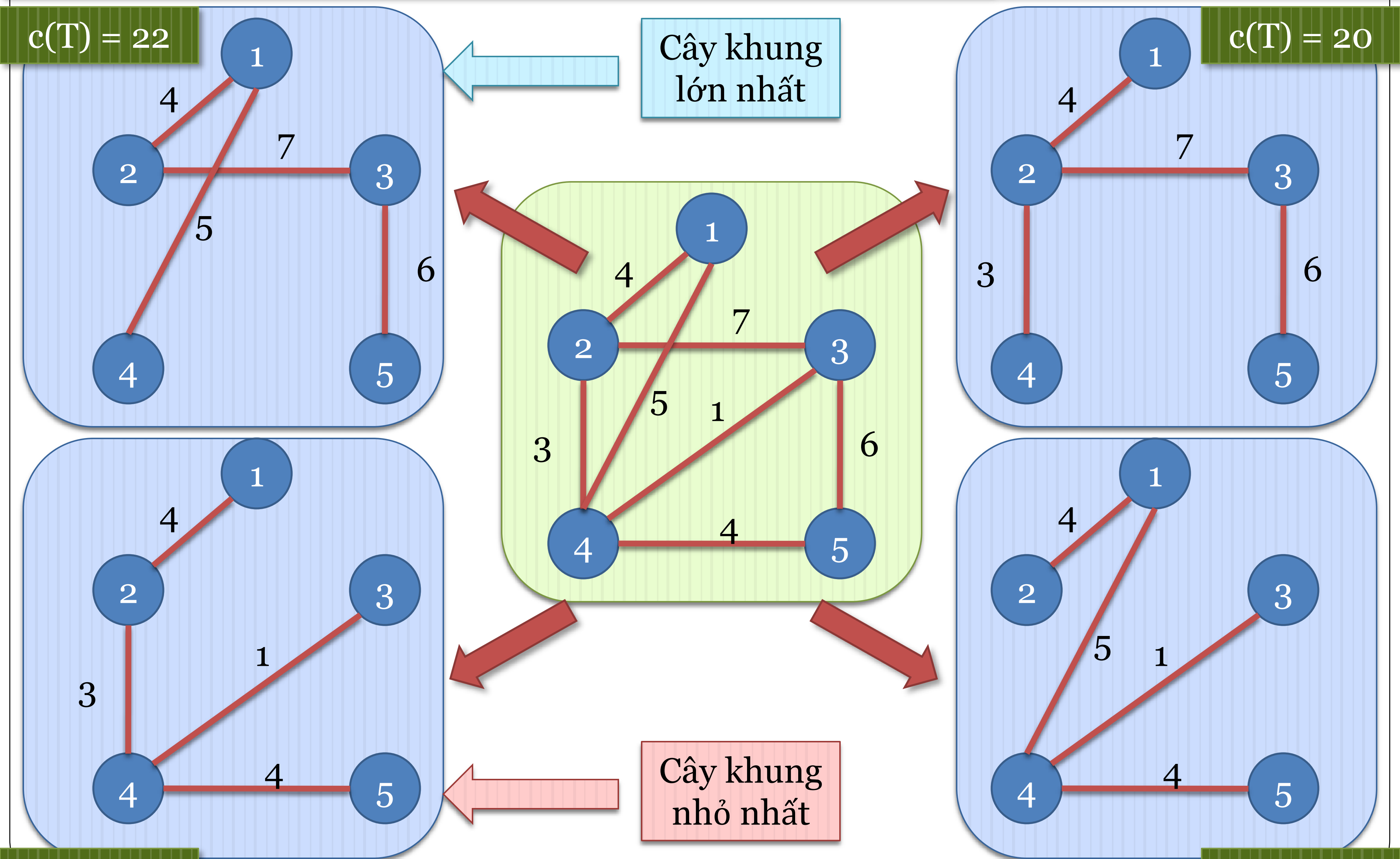
Ví dụ về độ dài cây khung



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 Kruscal do *Joseph Kruskal* phát minh năm 1956.

Kruskal (1928 -)

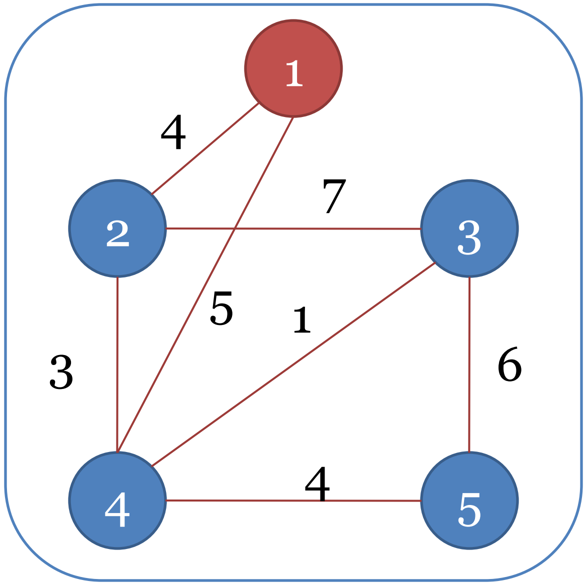
Prim (1921 -)

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 $u \in V$ và khởi tạo: $Y = \{u\}$ và $E_T = \varnothing$.
- **Bước 2.** Trong số những cạnh $e = \{u, v\}$, trong đó $u \in Y$ và $v \in V \backslash Y$, ta chọn cạnh $e_i = \{u_i, v_i\}$ có độ dài nhỏ nhất.
- **Bước 3.** Gán $Y = Y \cup \{v_i\}$ và $E_T = E_T \cup \{e_i\}$
- **Bước 4.** Nếu E_T đủ $n – 1$ phần tử thì dừng, ngược lại làm tiếp bước 2.
- *$T = (V, E_T)$ chính là cây khung nhỏ nhất.*

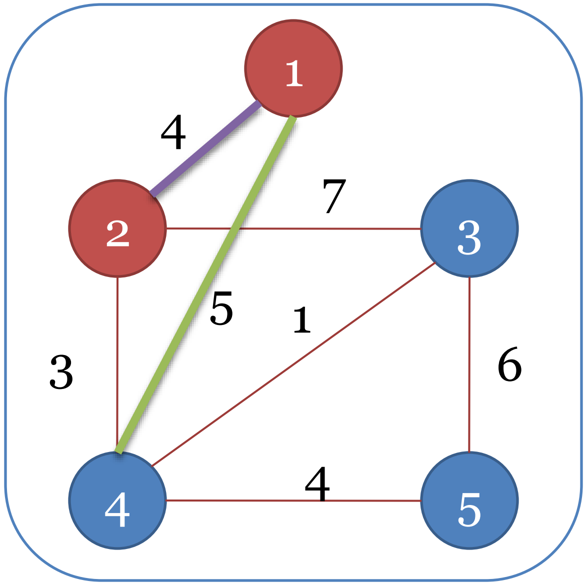
Ví dụ thuật toán Prim

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



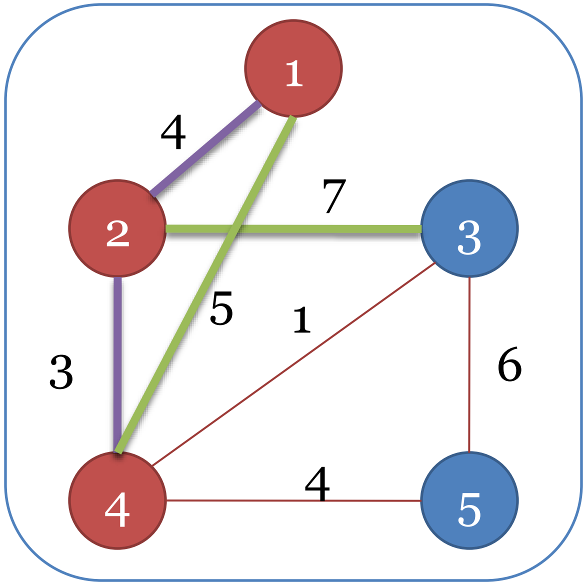
Ví dụ thuật toán Prim

- 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}
  - E_T = {{1, 2}}



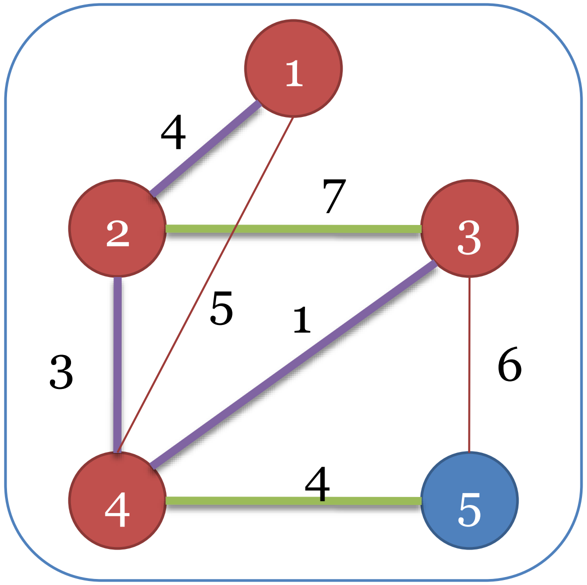
Ví dụ thuật toán Prim

- 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}
  - E_T = {{1, 2}, {2, 4}}



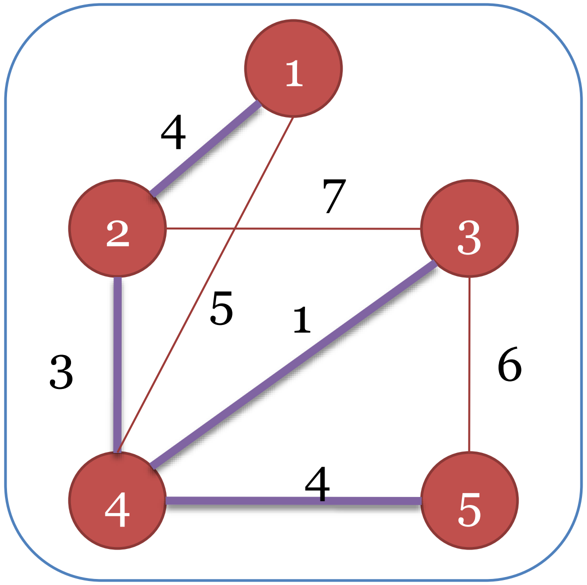
Ví dụ thuật toán Prim

- 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}
  - E_T = {{1, 2}, {2, 4}, {4, 3}}



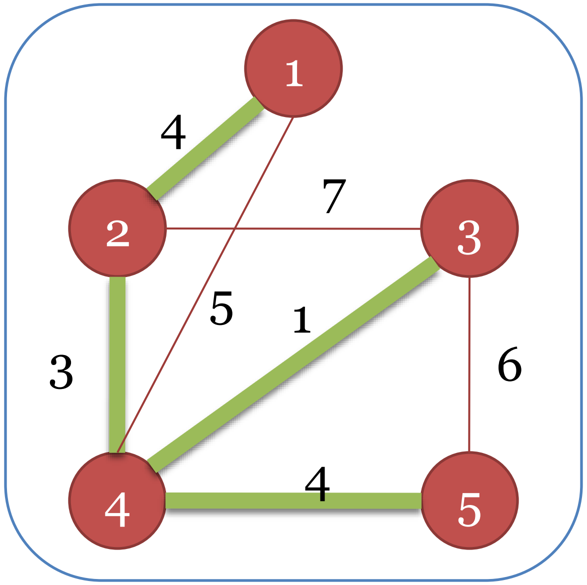
Ví dụ thuật toán Prim

- 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:
  - Y = {1, 2, 4, 3, 5}
  - E_T = {{1, 2}, {2, 4}, {4, 3}, {4, 5}}



Ví dụ thuật toán Prim

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

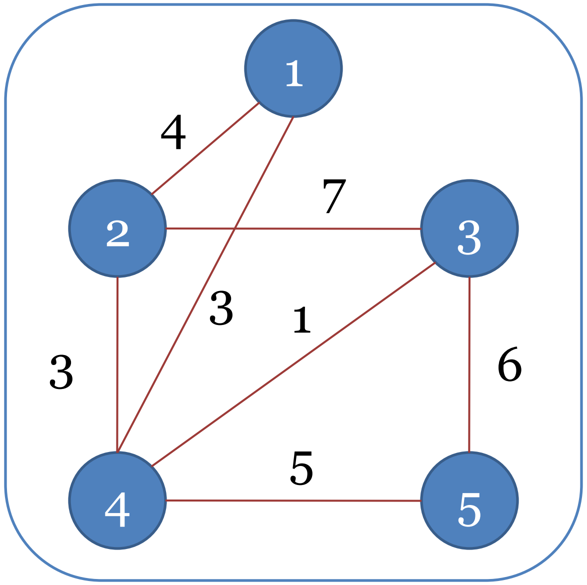


Thuật toán Kruskal

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

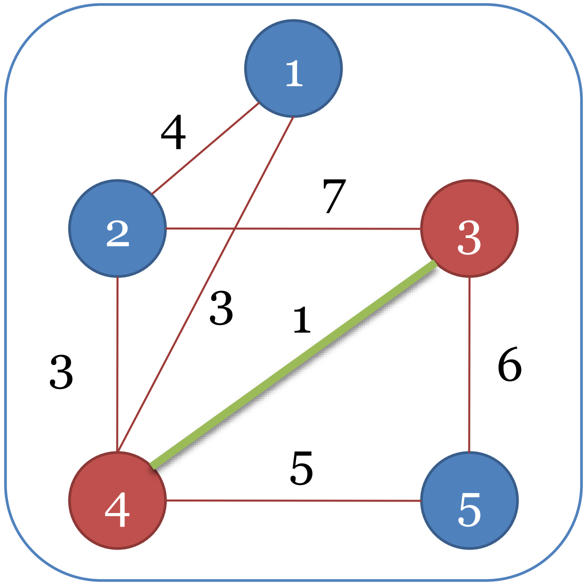
Ví dụ thuật toán Kruskal

- 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
- $E_T = \varnothing$



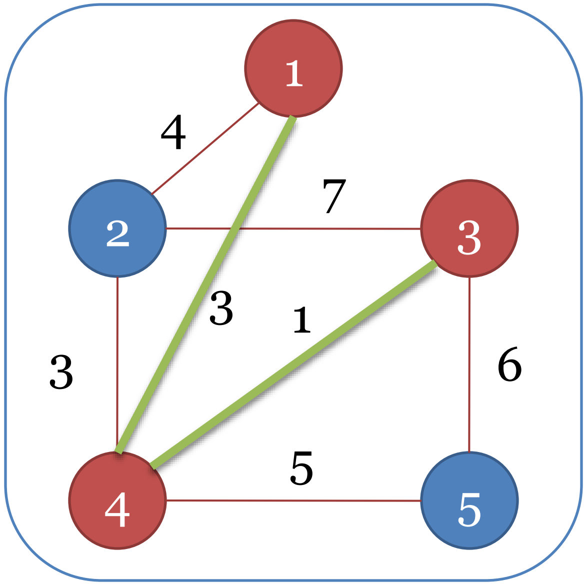
Ví dụ thuật toán Kruskal

- Chọn cạnh {3, 4} – là cạnh có trọng nhỏ nhất.
- 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



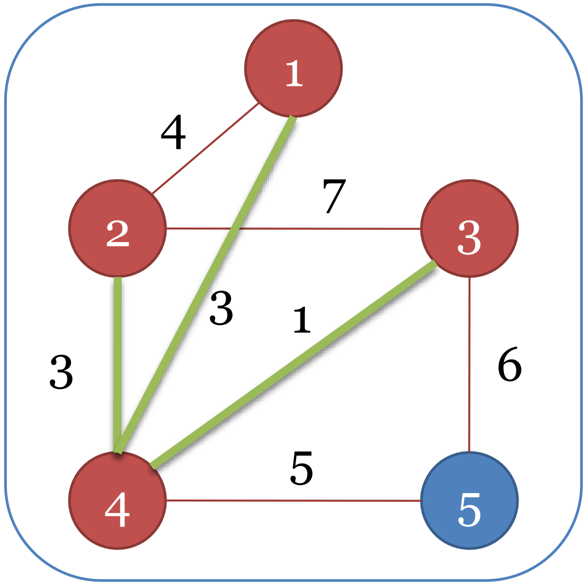
Ví dụ thuật toán Kruskal

- Chọn cạnh {1, 4} – là cạnh tiếp theo trong danh sách.
- 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



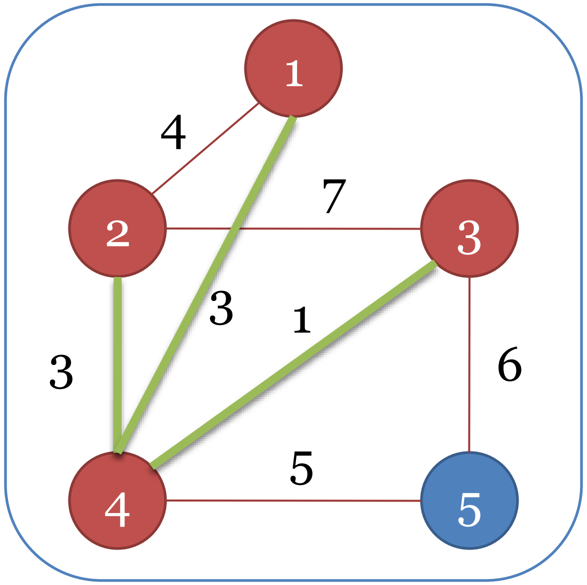
Ví dụ thuật toán Kruskal

- Chọn cạnh {2, 4} – là cạnh tiếp theo trong danh sách.
- 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



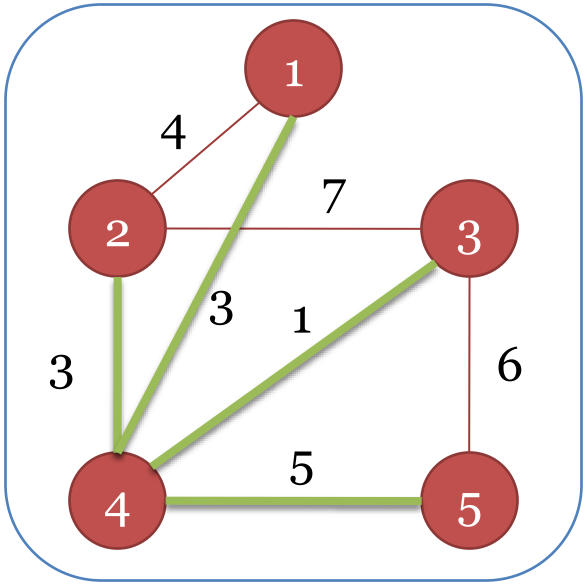
Ví dụ thuật toán Kruskal

- KHÔNG chọn cạnh {1, 2} – là cạnh tiếp theo trong danh sách vì khi chọn sẽ tạo thành chu trình.
- 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



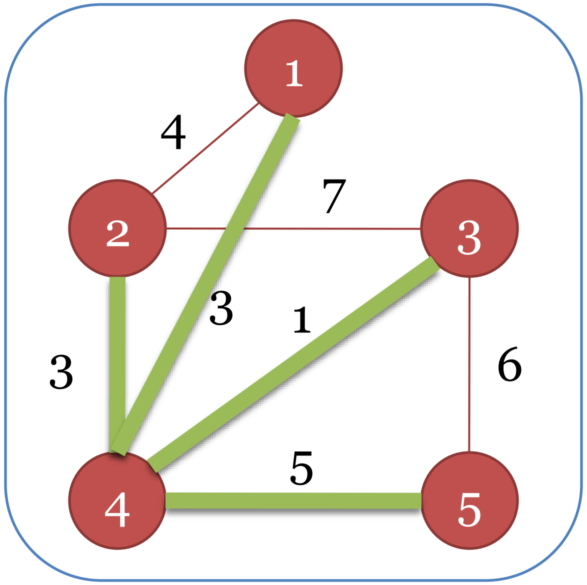
Ví dụ thuật toán Kruskal

- Chọn cạnh {4, 5} – là cạnh tiếp theo trong danh sách.
- 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 thuật toán Kruskal

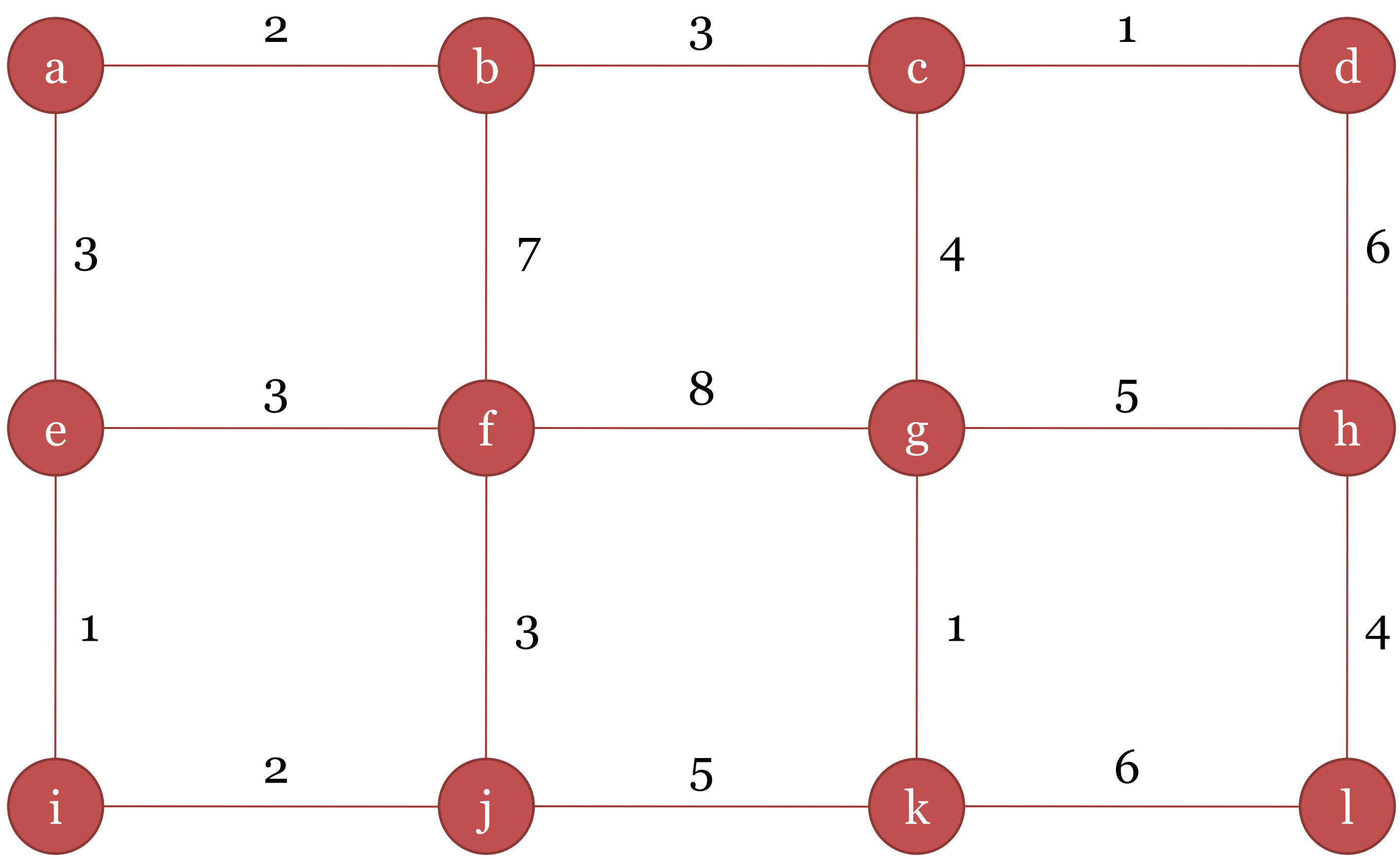
- Do $|E_T| = 4$, kết thúc thuật toán.
- Cây khung tìm được là:
  - $E_T = \{\{1, 4\}, \{2, 4\}, \{3, 4\}, \{4, 5\}\}$



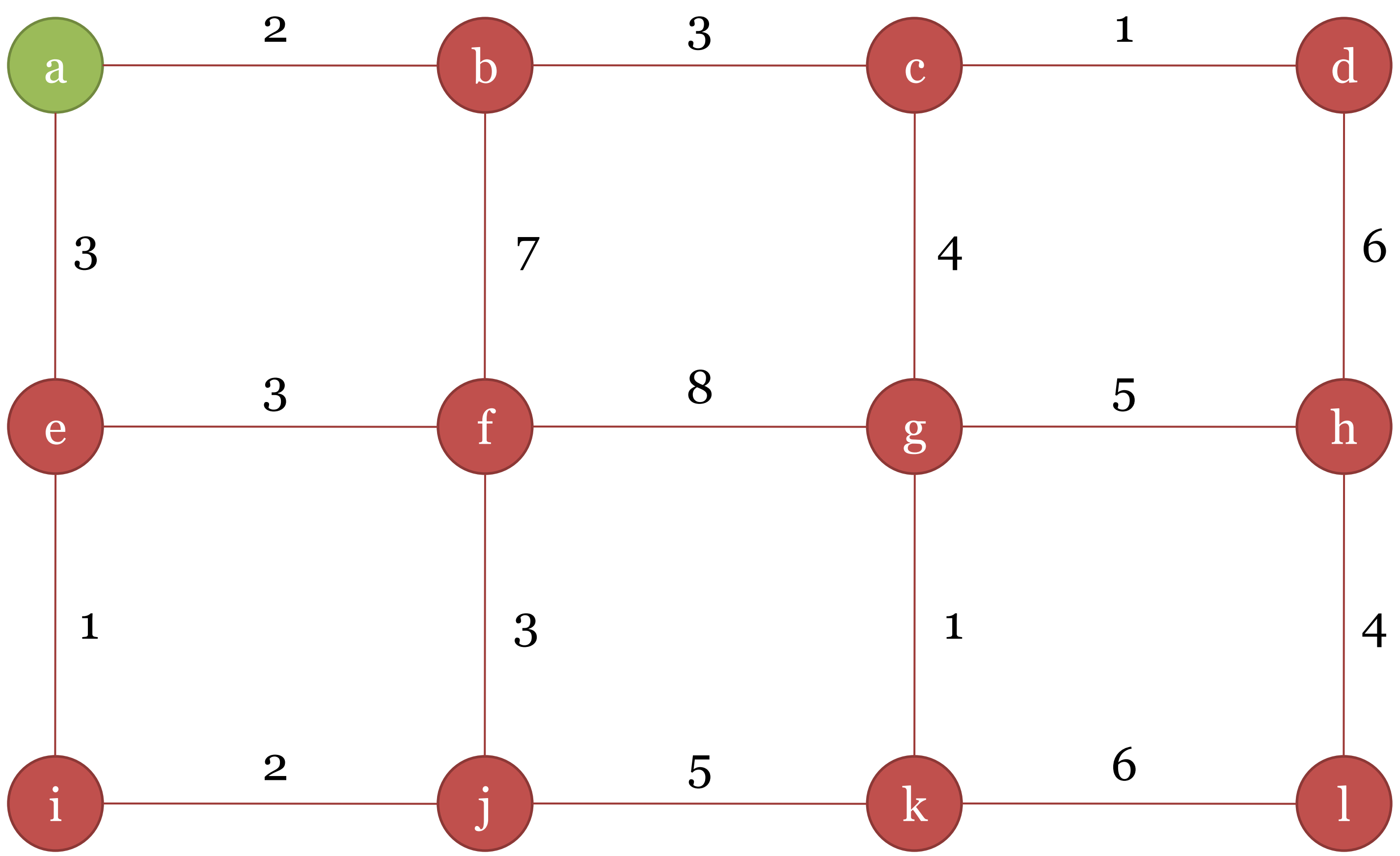
Tìm cây khung lớn nhất

- 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.

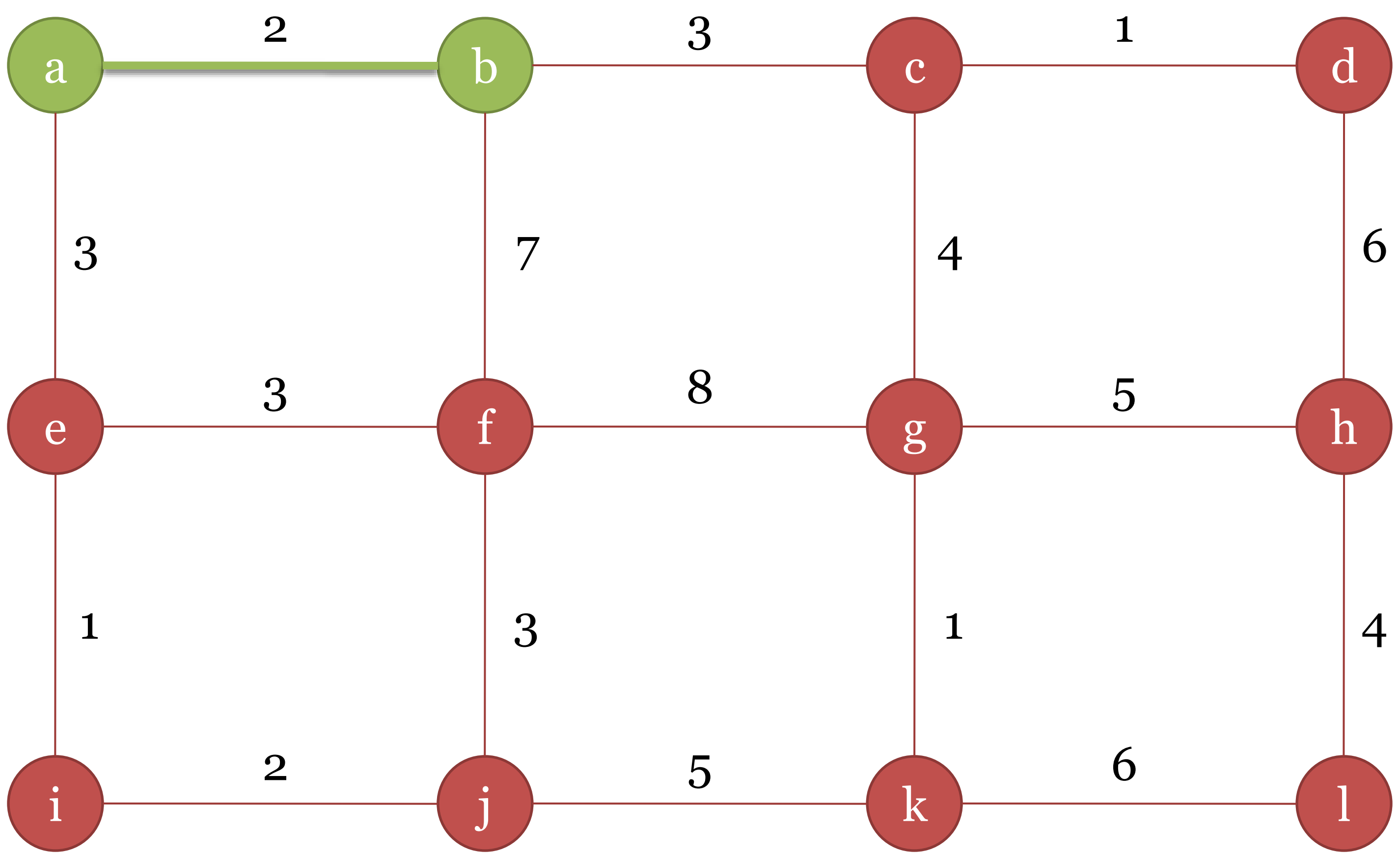
Ví dụ tìm cây khung nhỏ nhất



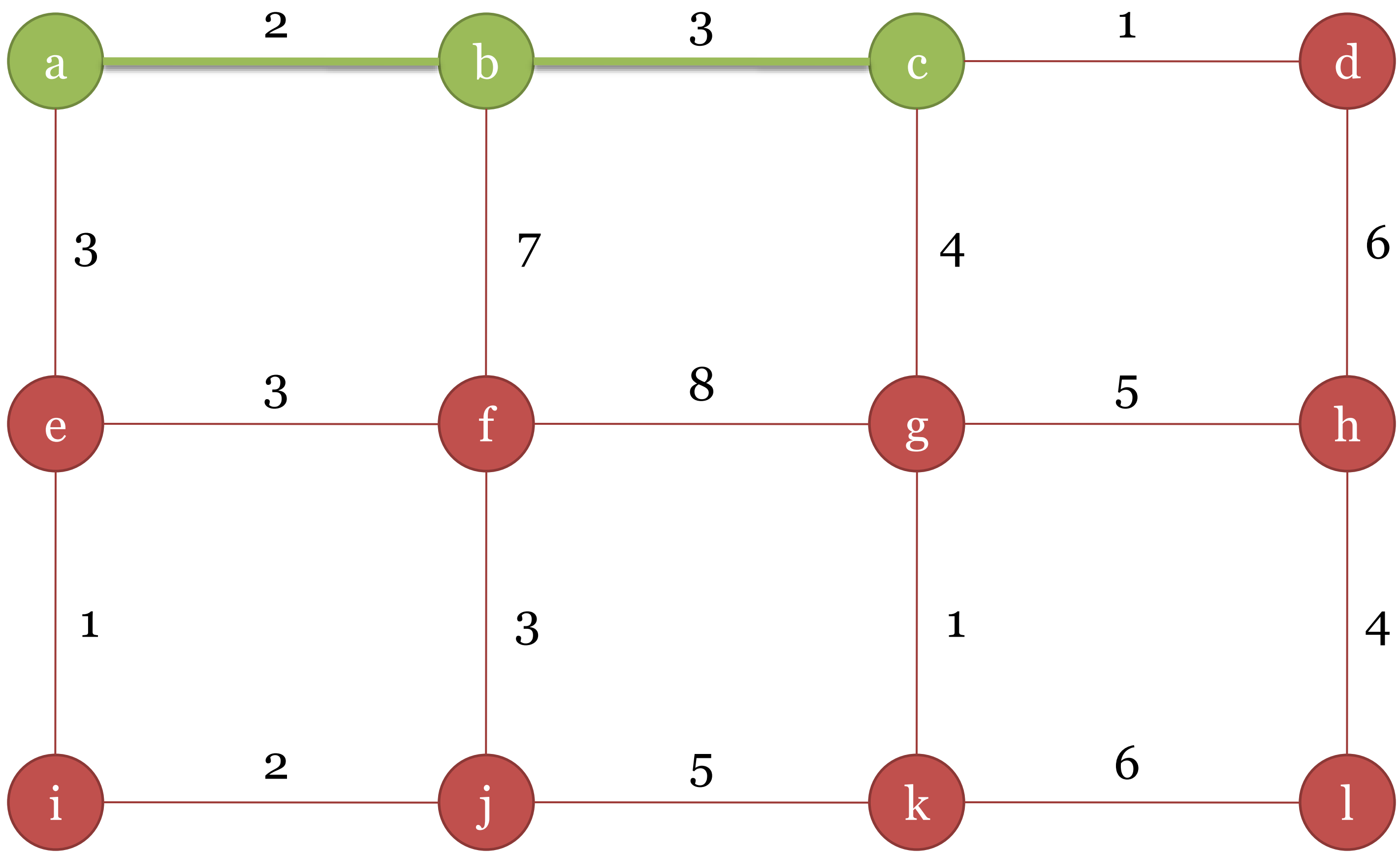
Tìm cây khung nhỏ nhất theo Prim



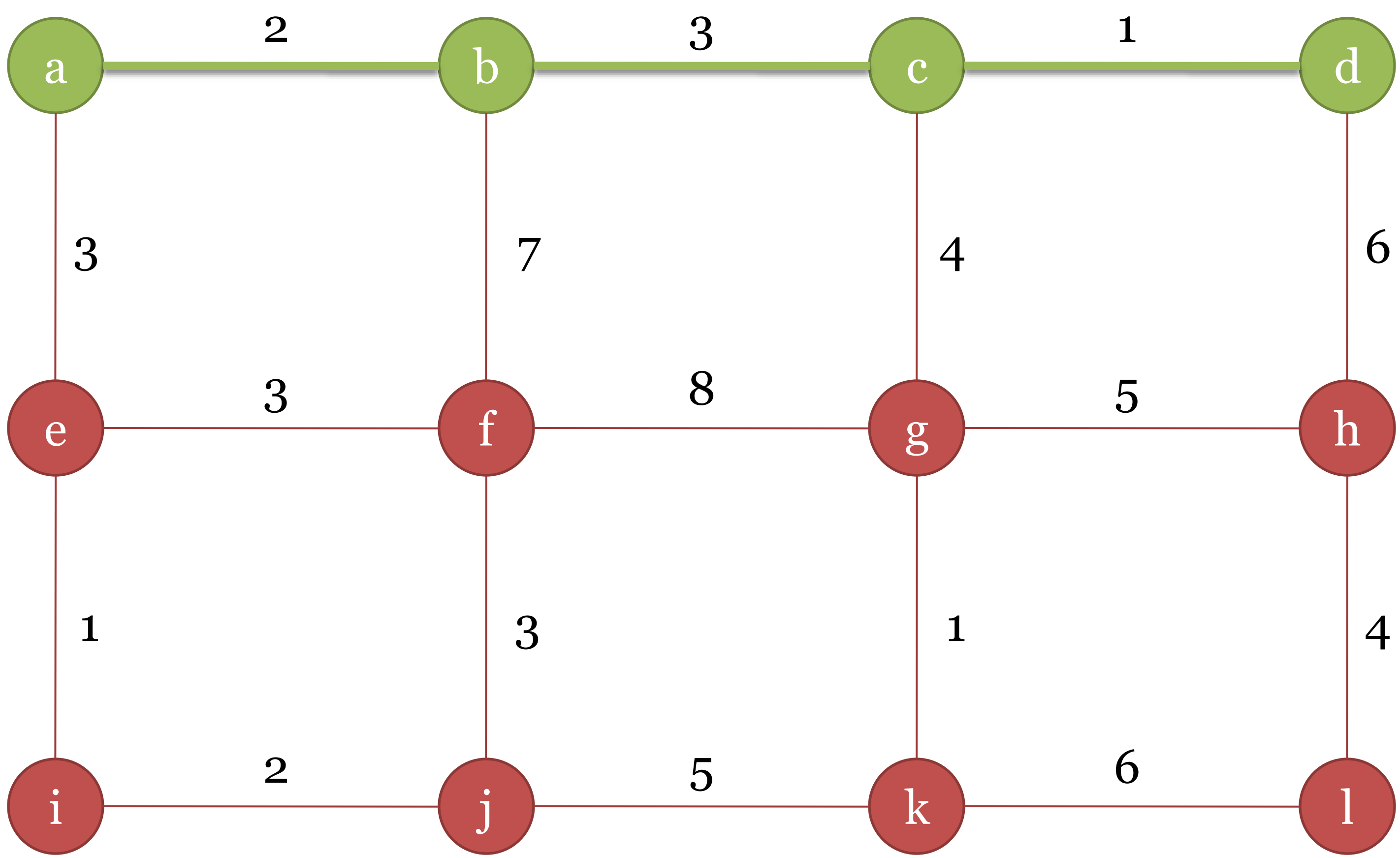
Tìm cây khung nhỏ nhất theo Prim



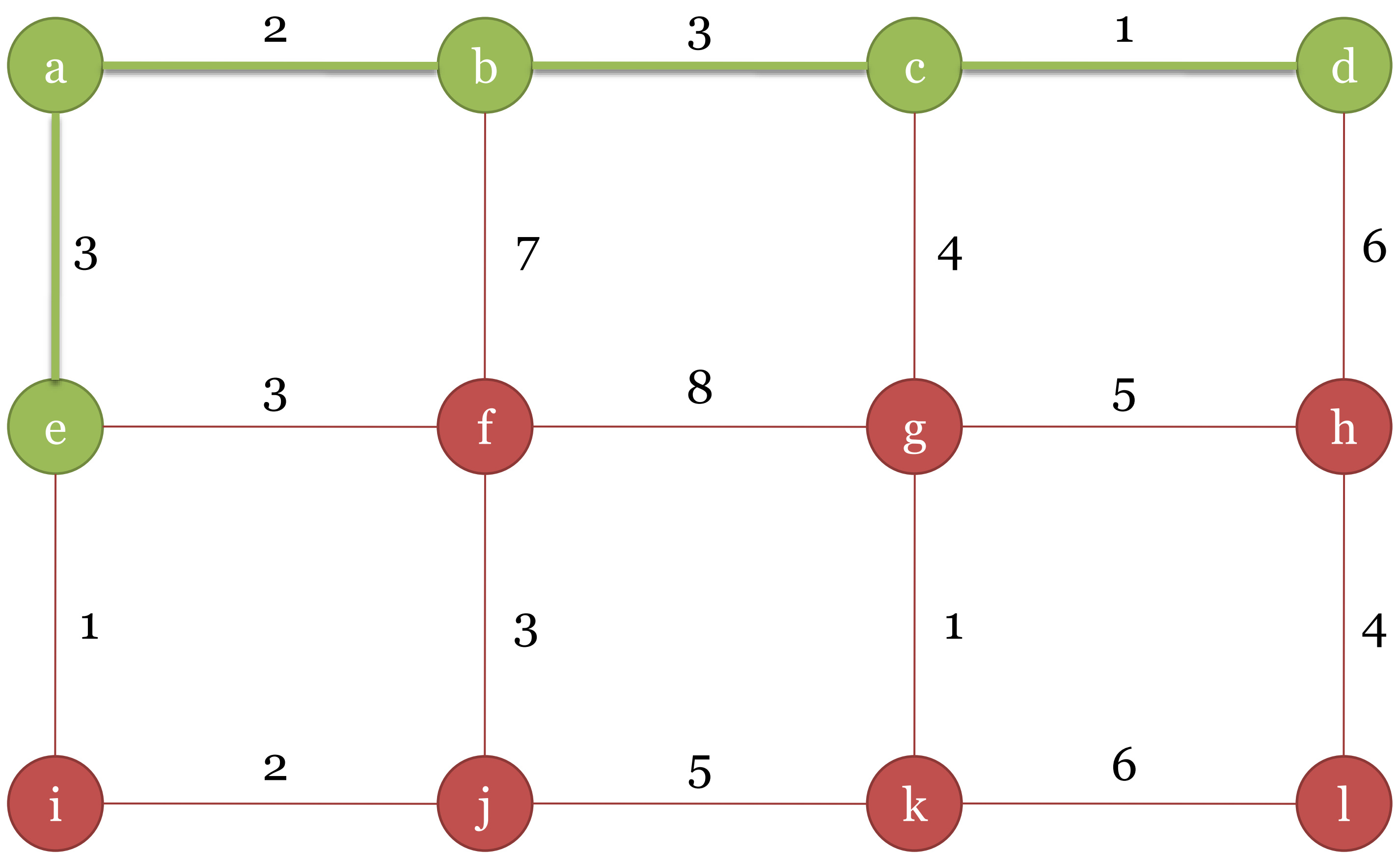
Tìm cây khung nhỏ nhất theo Prim



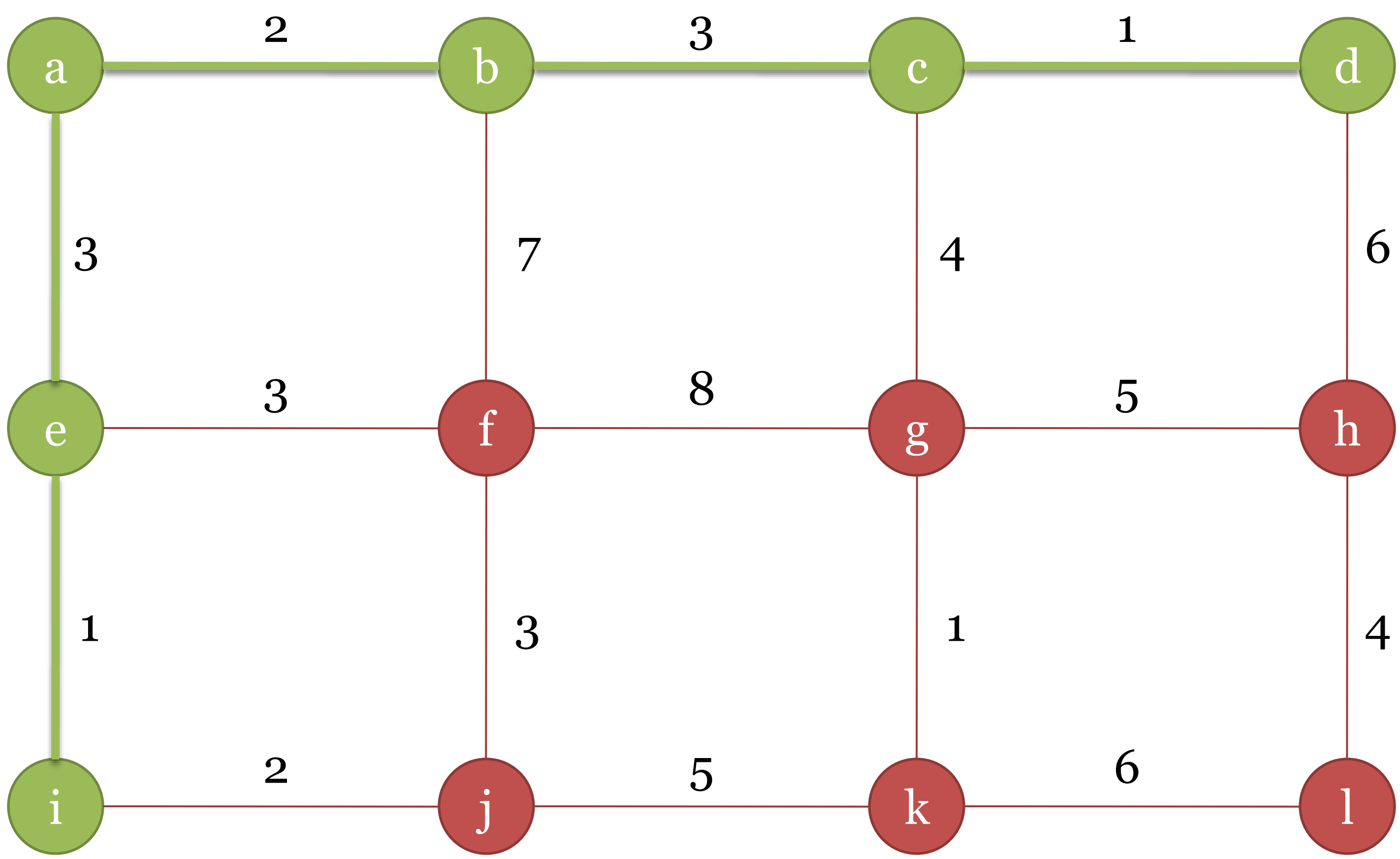
Tìm cây khung nhỏ nhất theo Prim



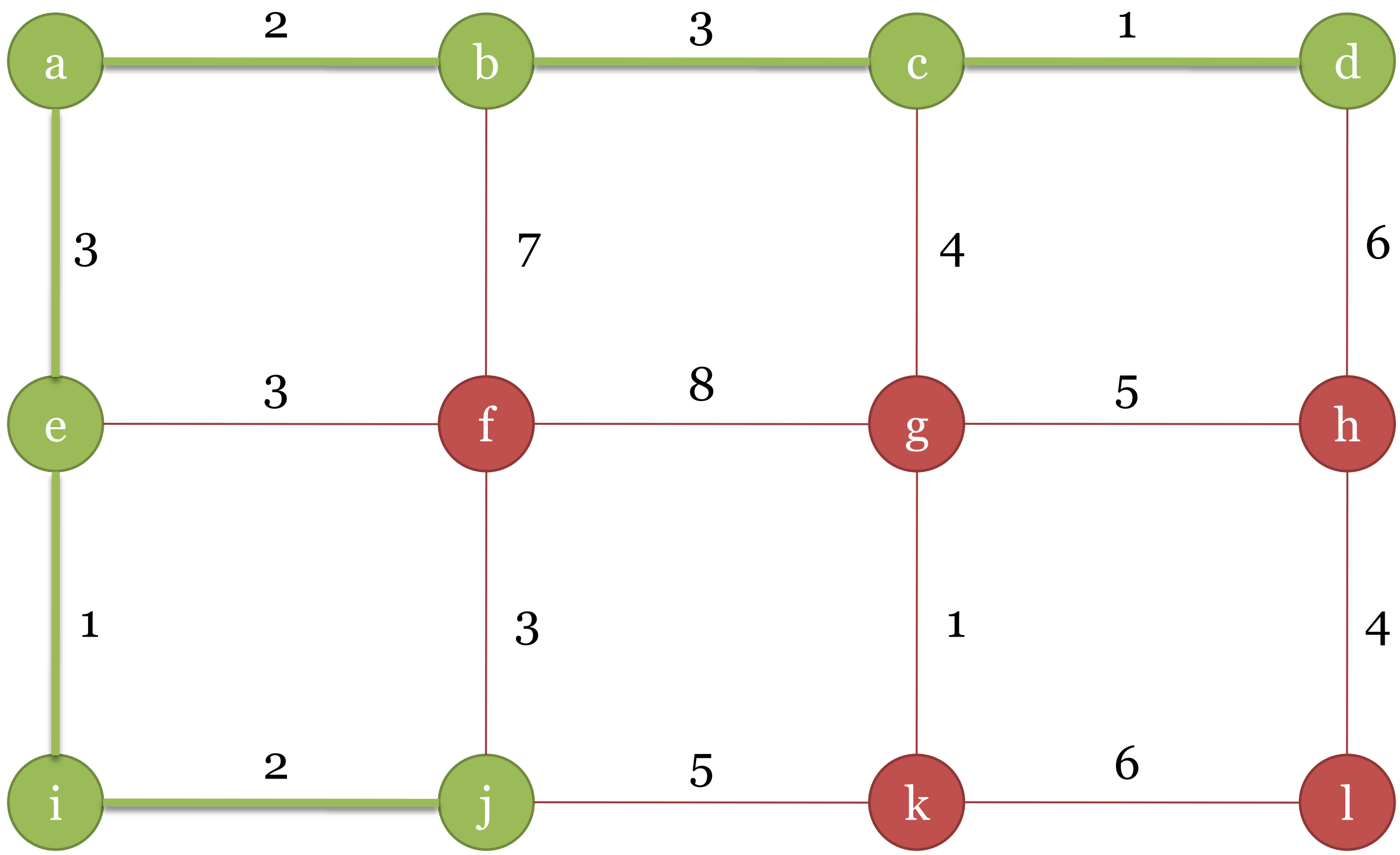
Tìm cây khung nhỏ nhất theo Prim



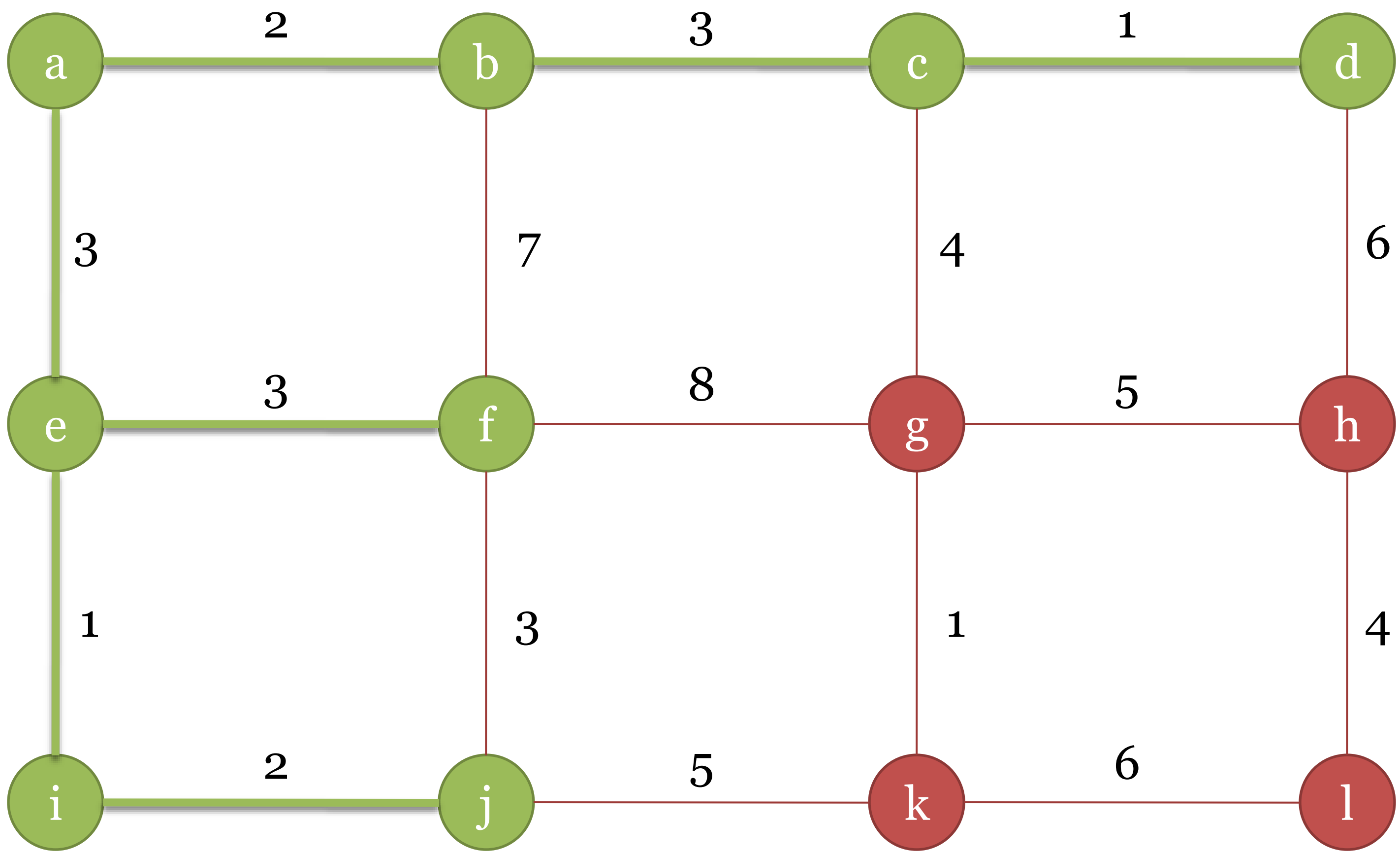
Tìm cây khung nhỏ nhất theo Prim



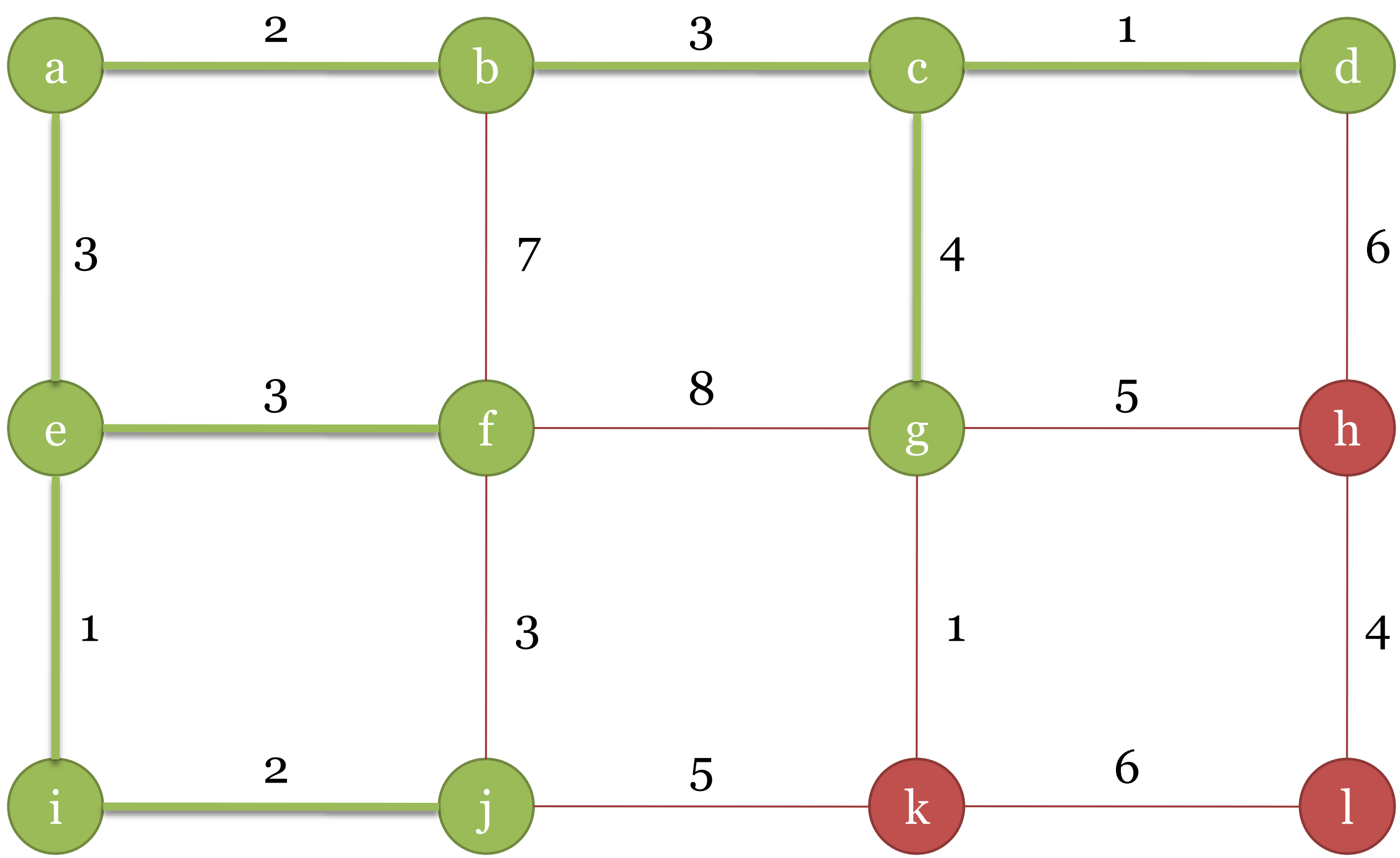
Tìm cây khung nhỏ nhất theo Prim



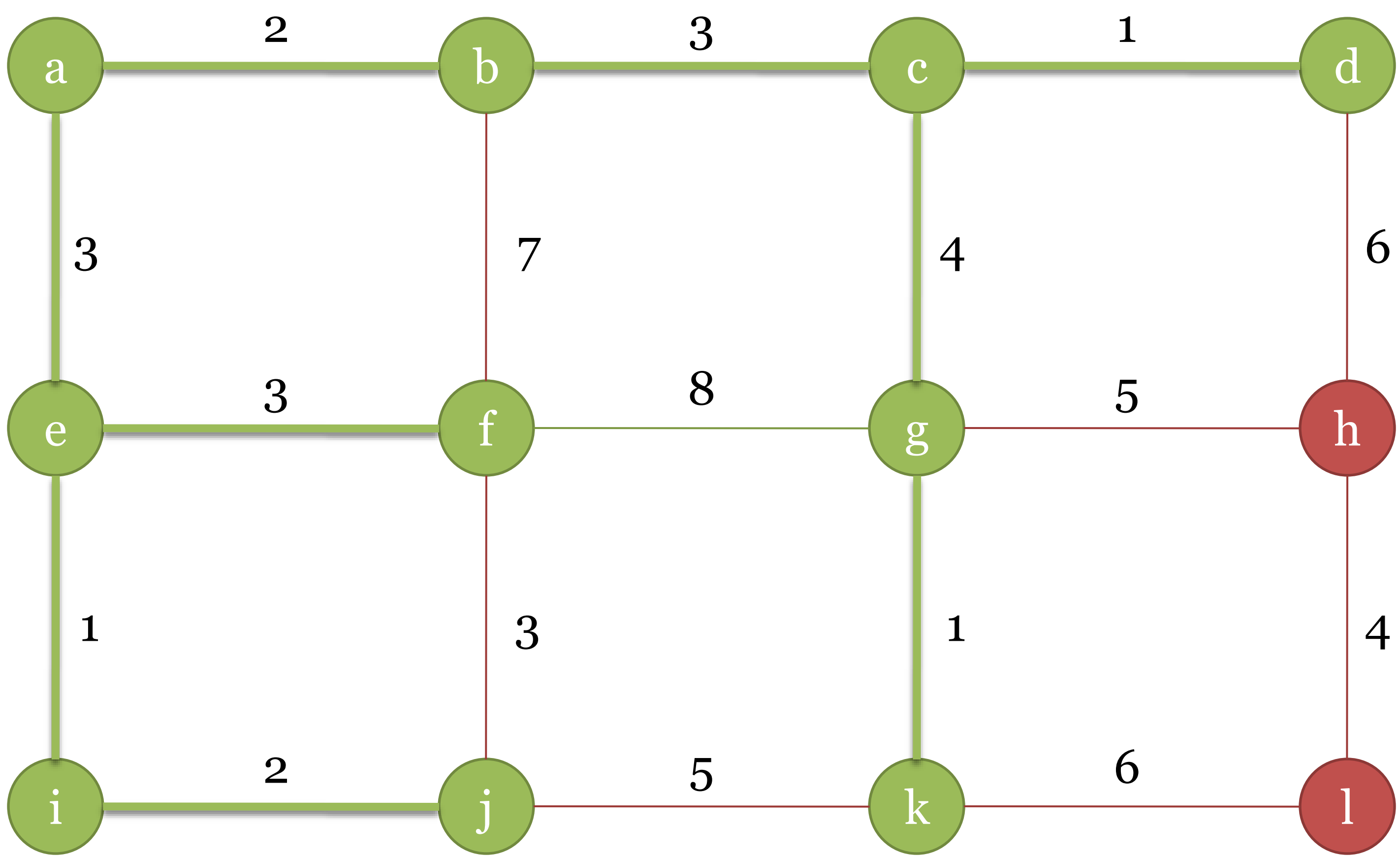
Tìm cây khung nhỏ nhất theo Prim



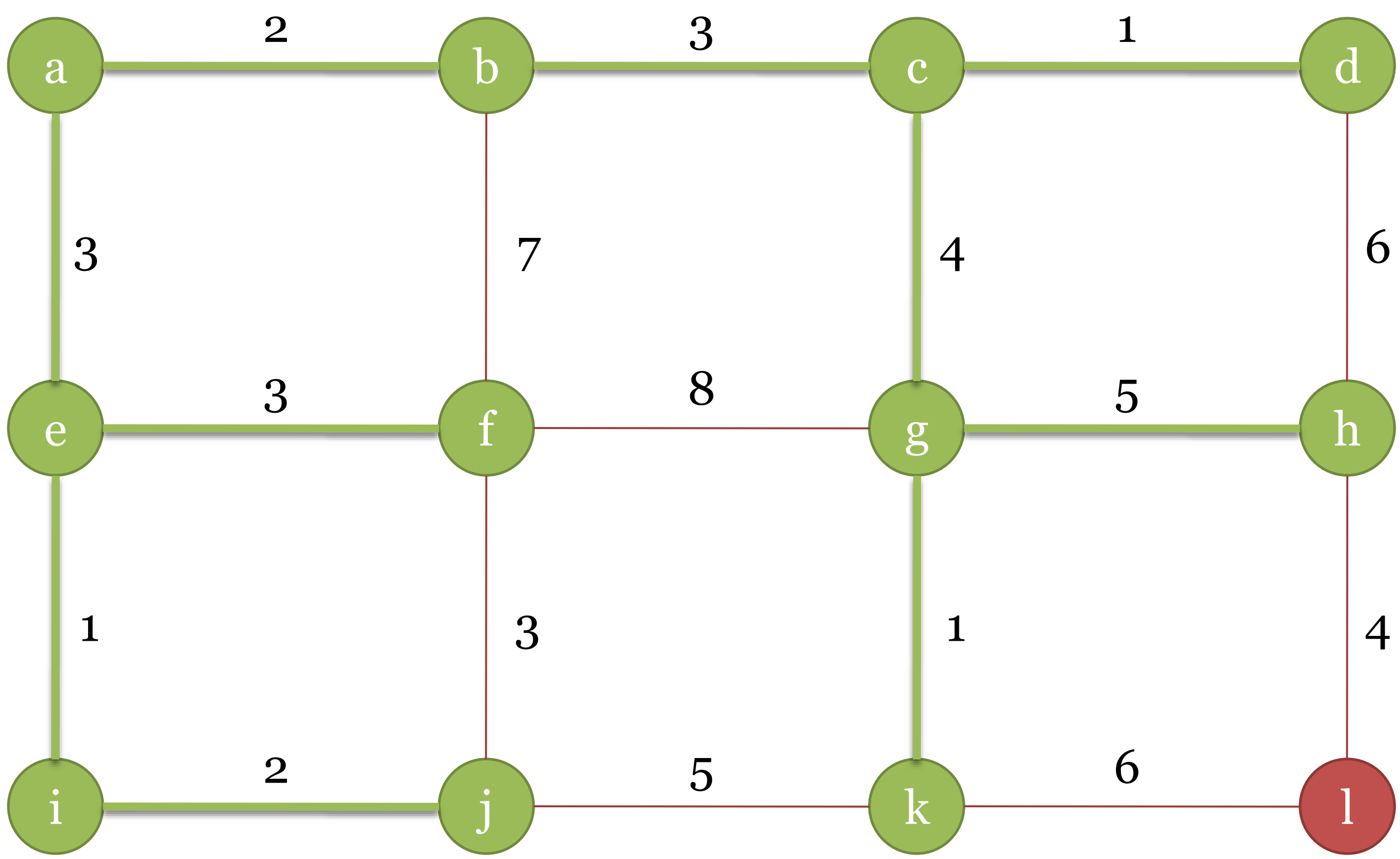
Tìm cây khung nhỏ nhất theo Prim



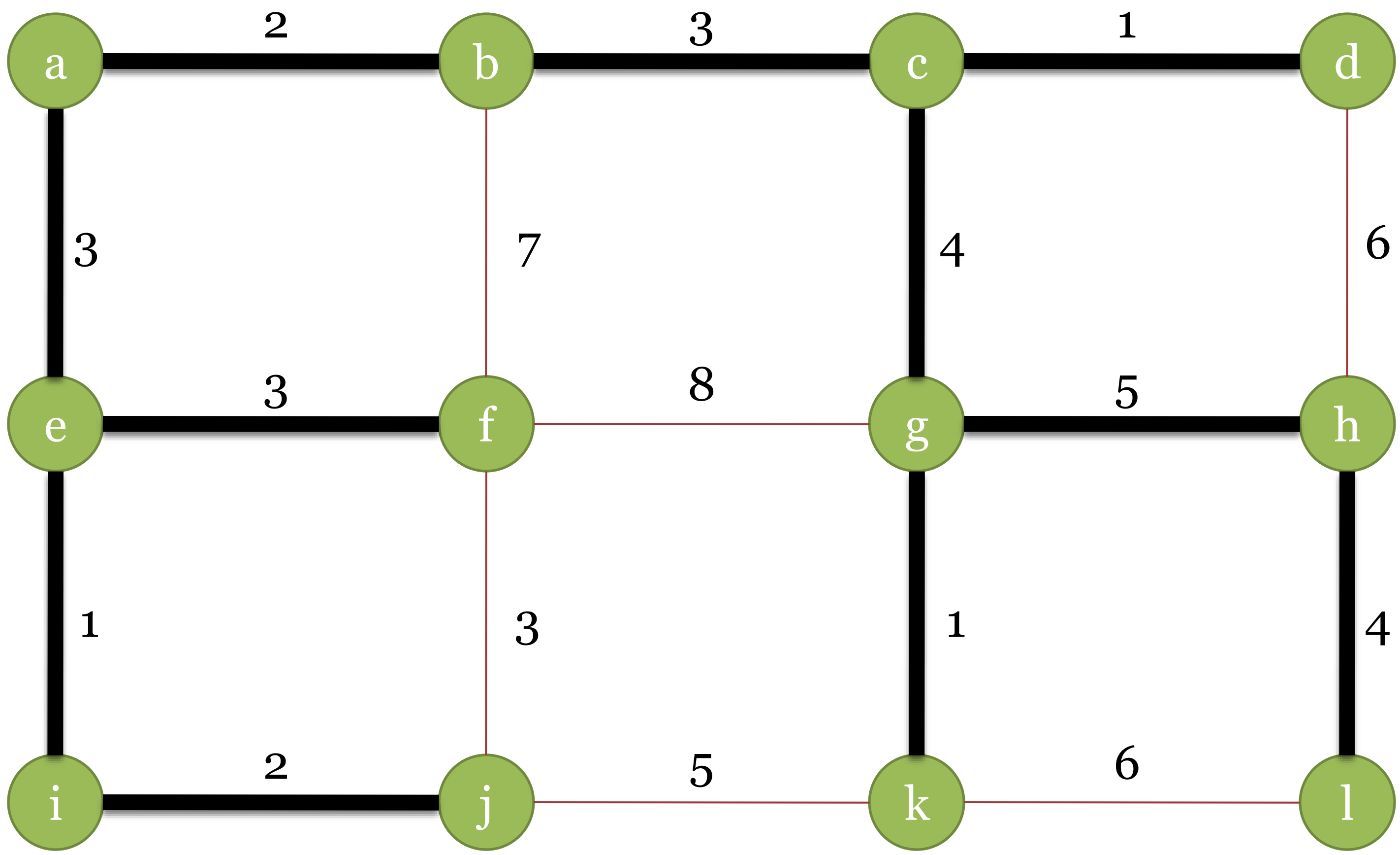
Tìm cây khung nhỏ nhất theo Prim



Tìm cây khung nhỏ nhất theo Prim

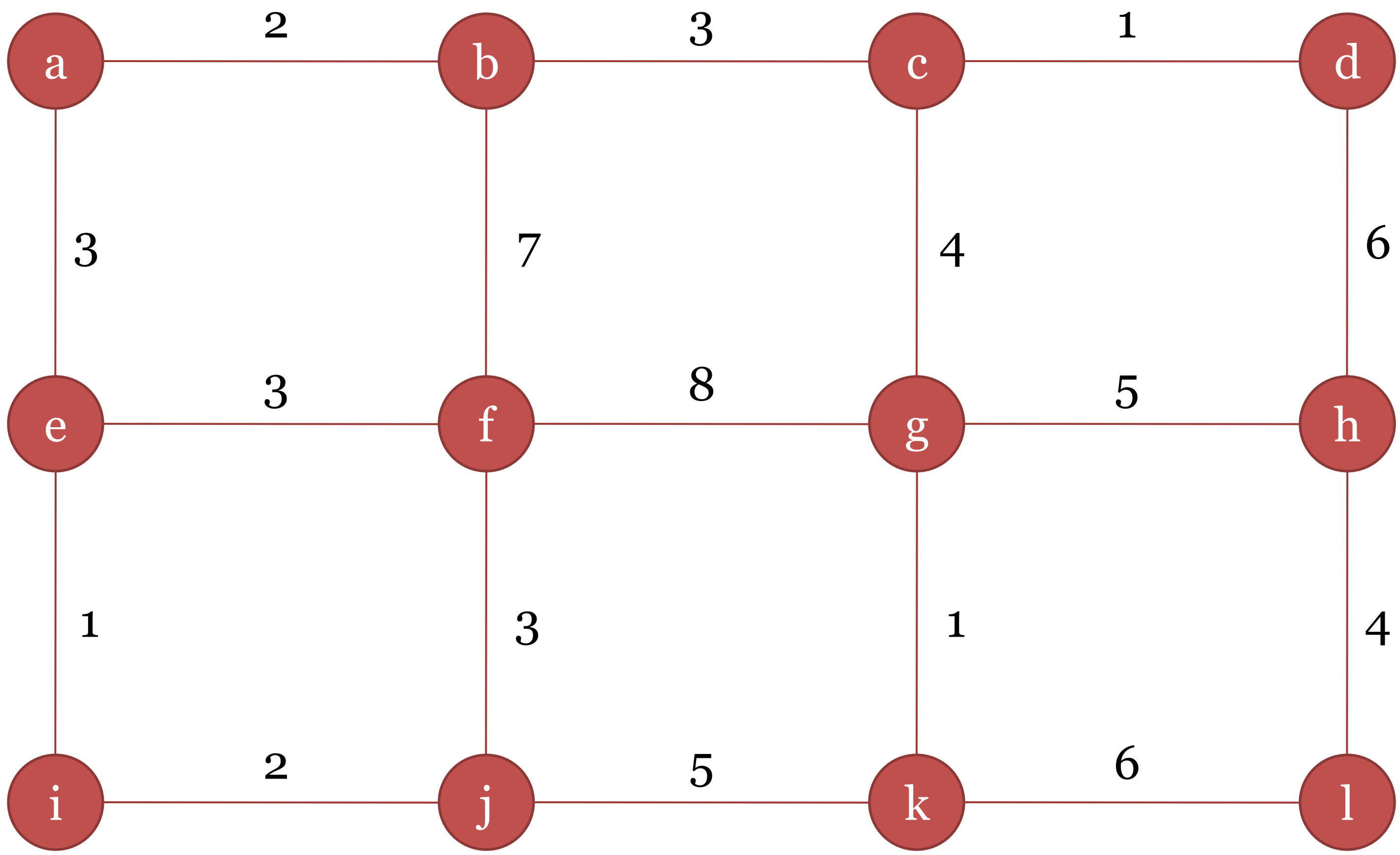


Tìm cây khung nhỏ nhất theo Prim

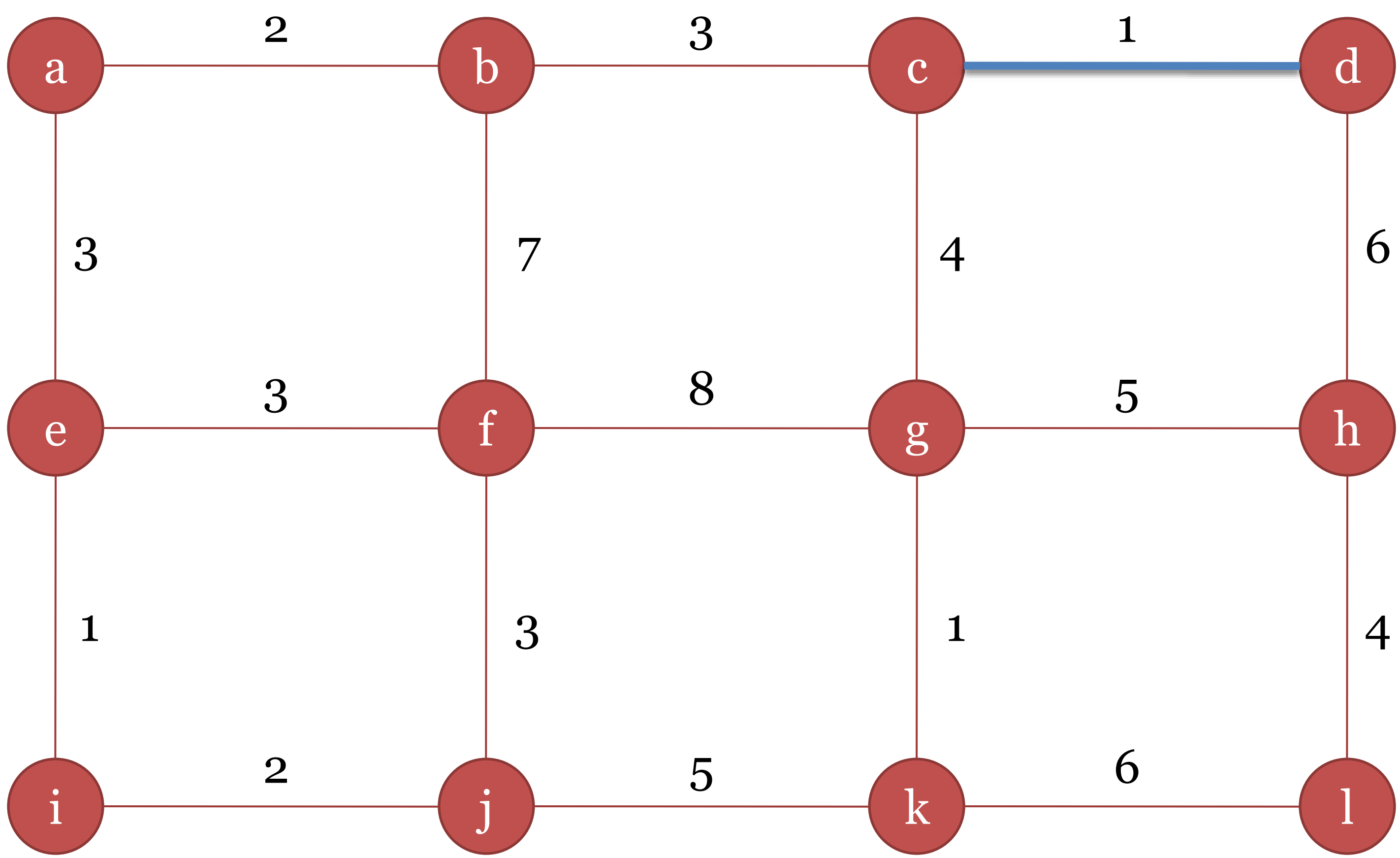


c(T) = 29

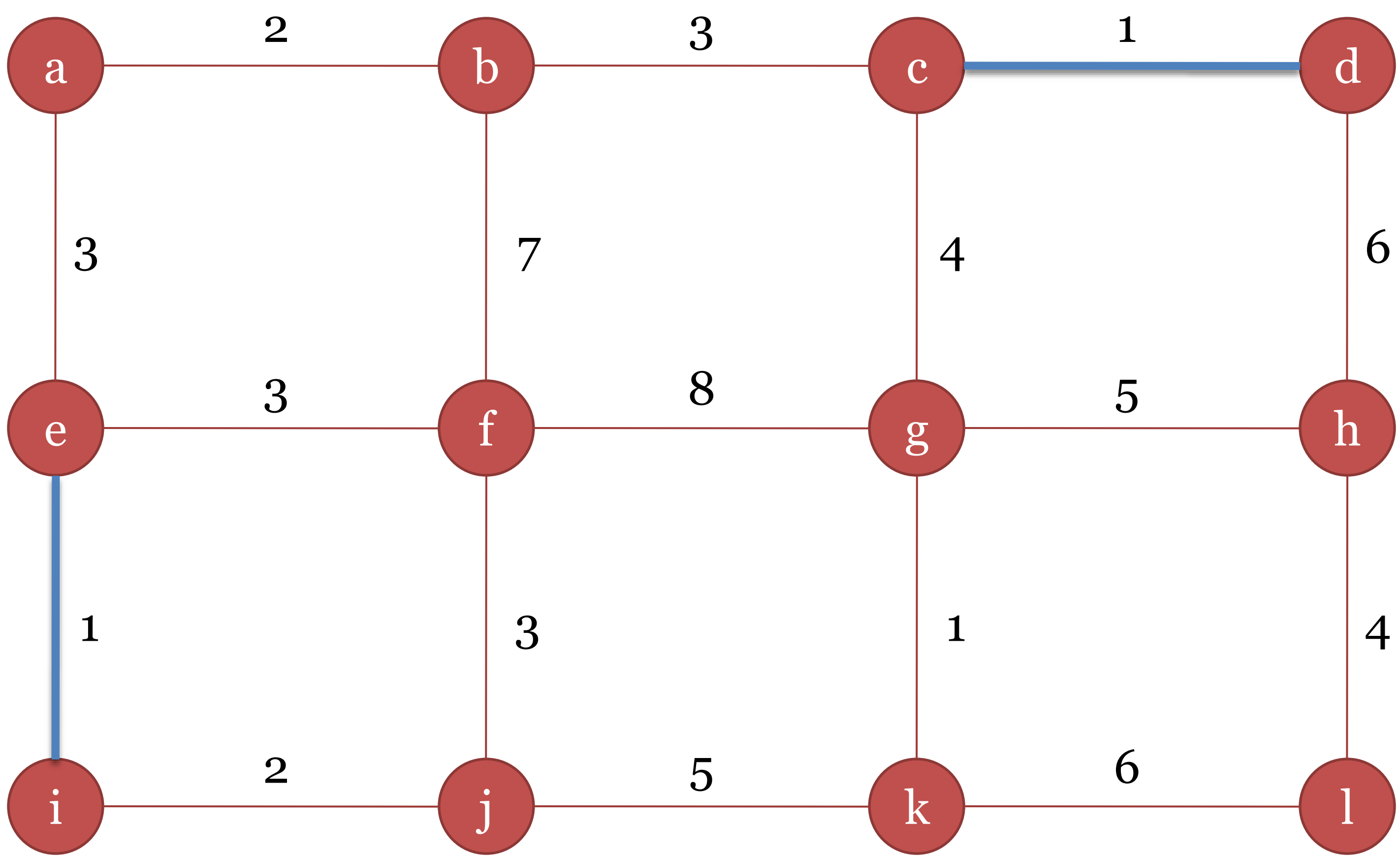
Tìm cây khung nhỏ nhất theo Kruskal



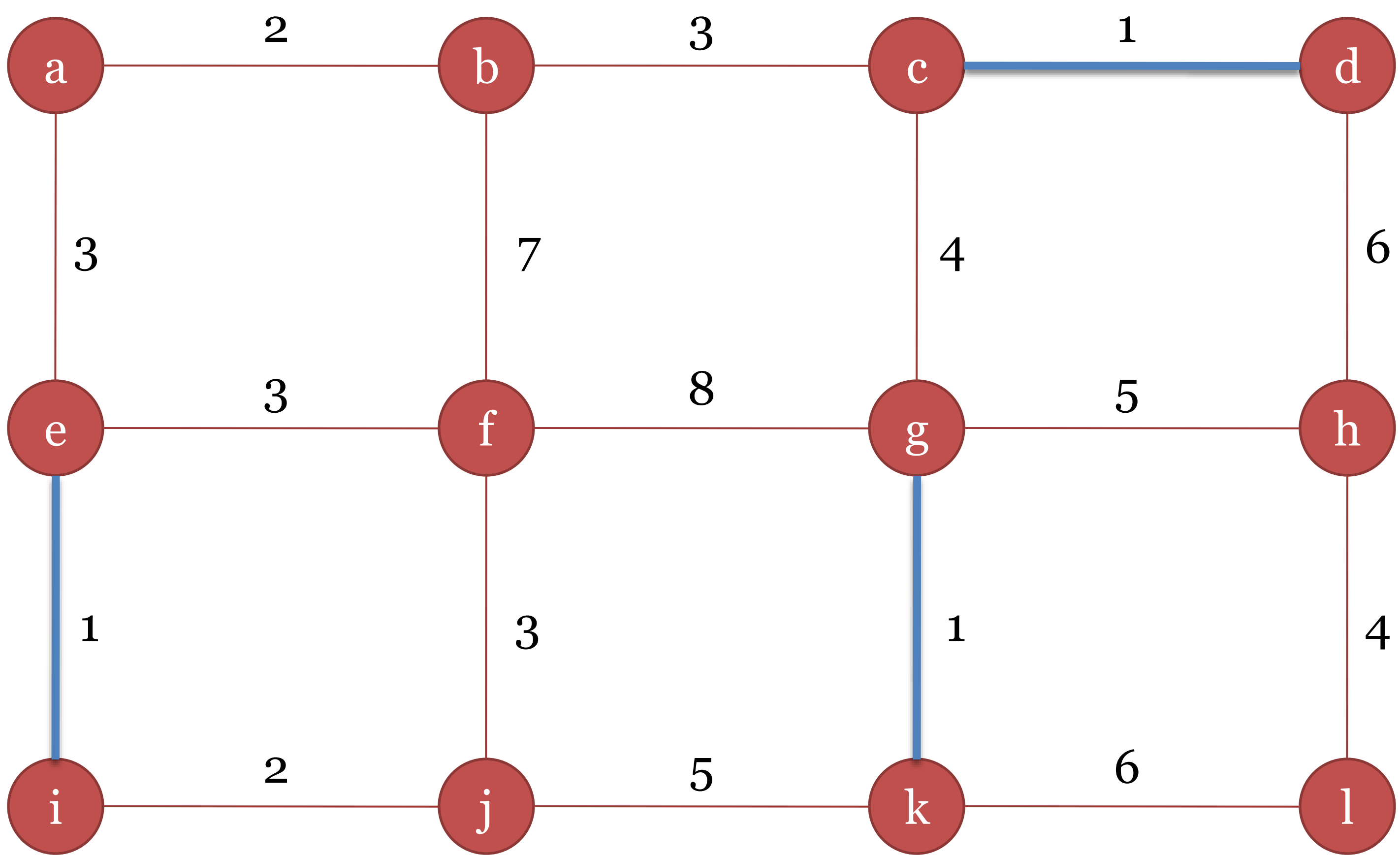
Tìm cây khung nhỏ nhất theo Kruskal



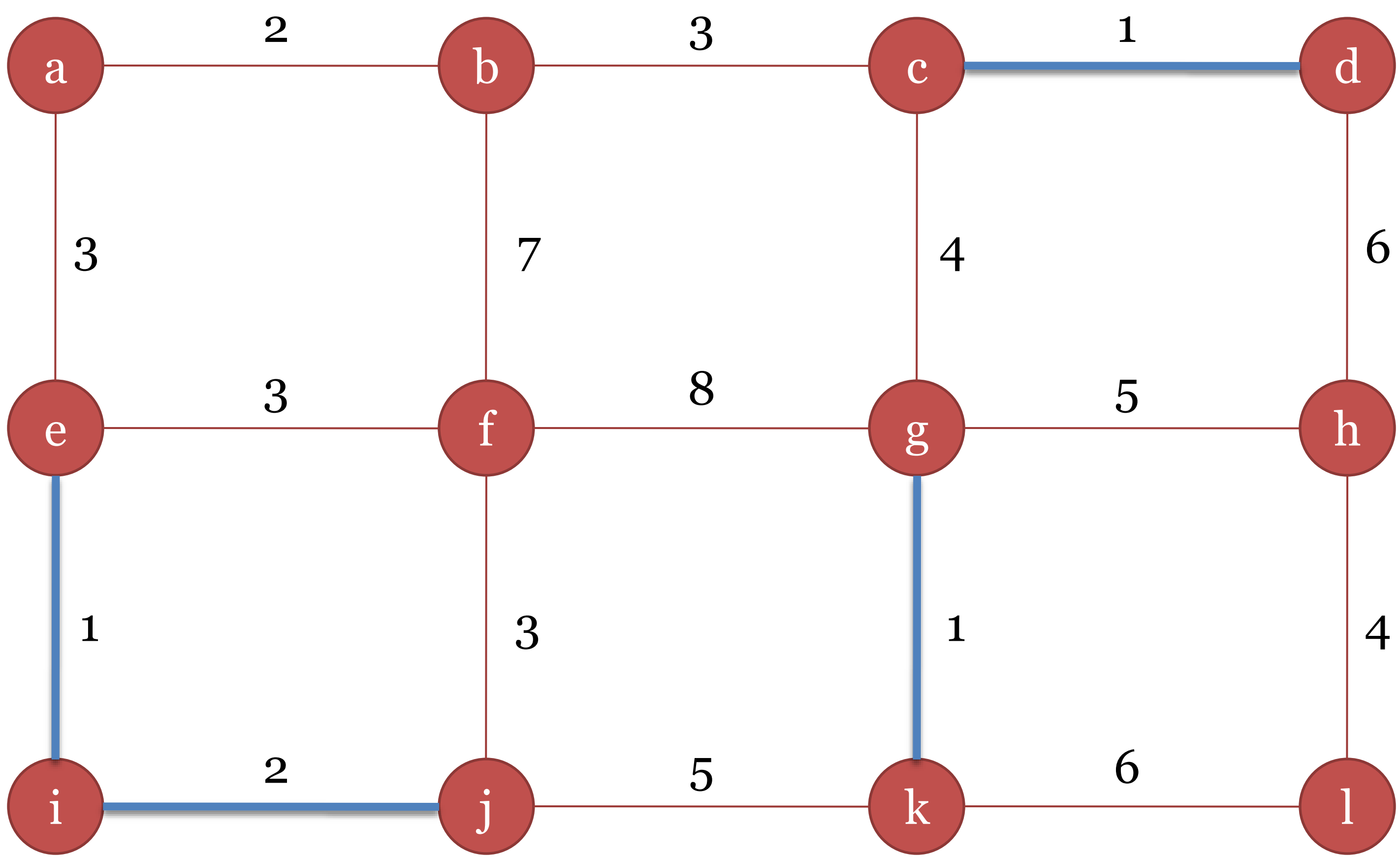
Tìm cây khung nhỏ nhất theo Kruskal



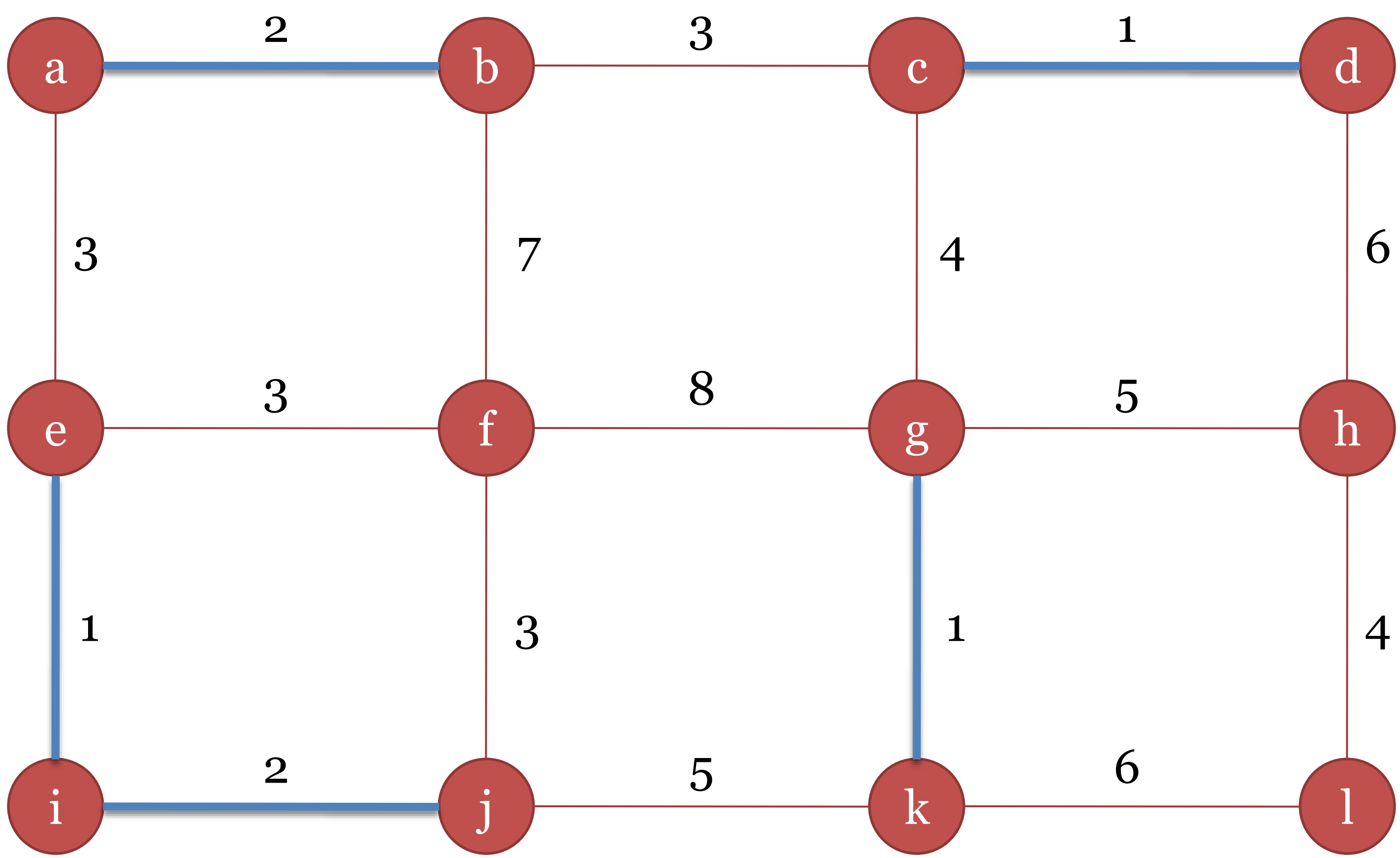
Tìm cây khung nhỏ nhất theo Kruskal



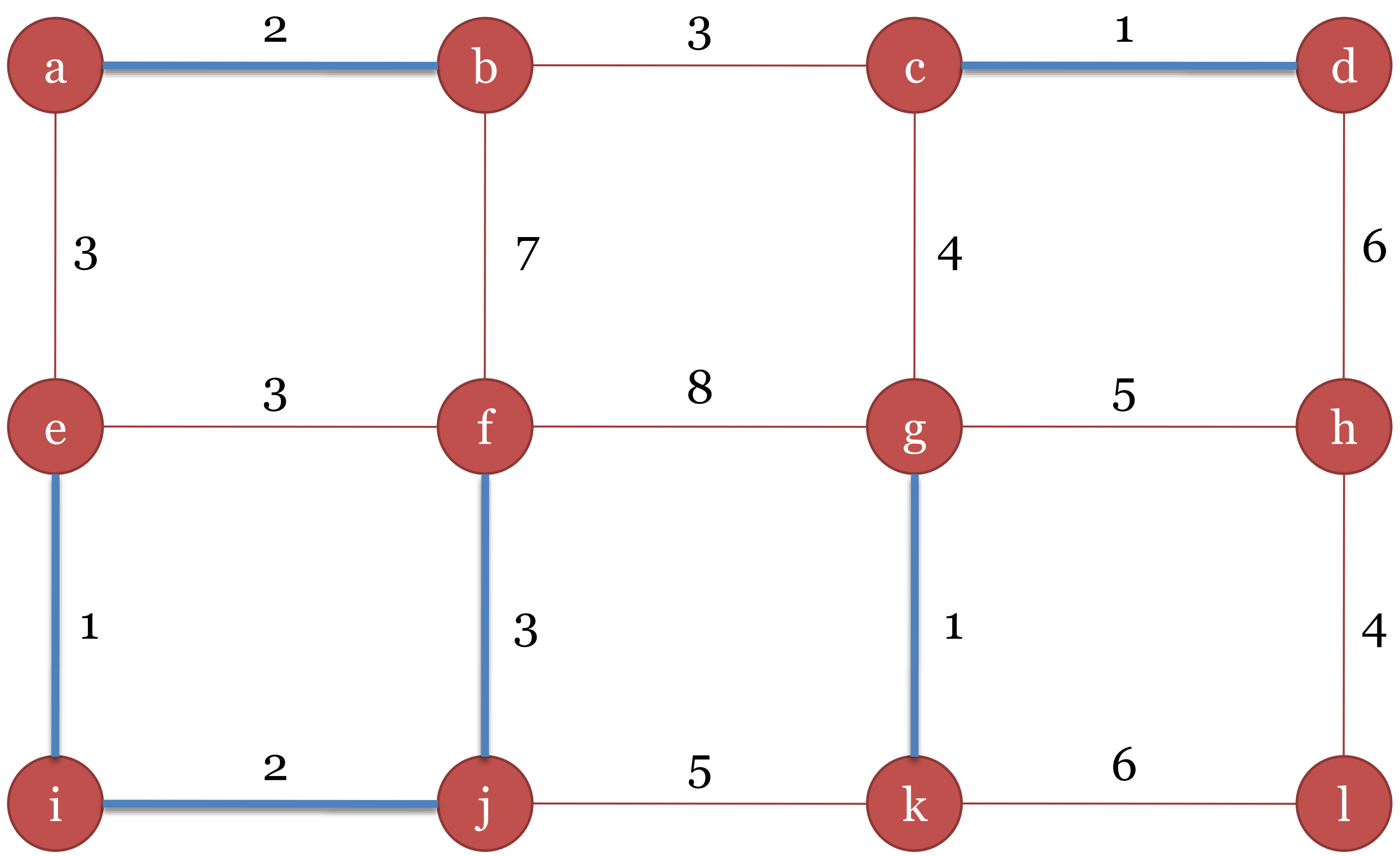
Tìm cây khung nhỏ nhất theo Kruskal



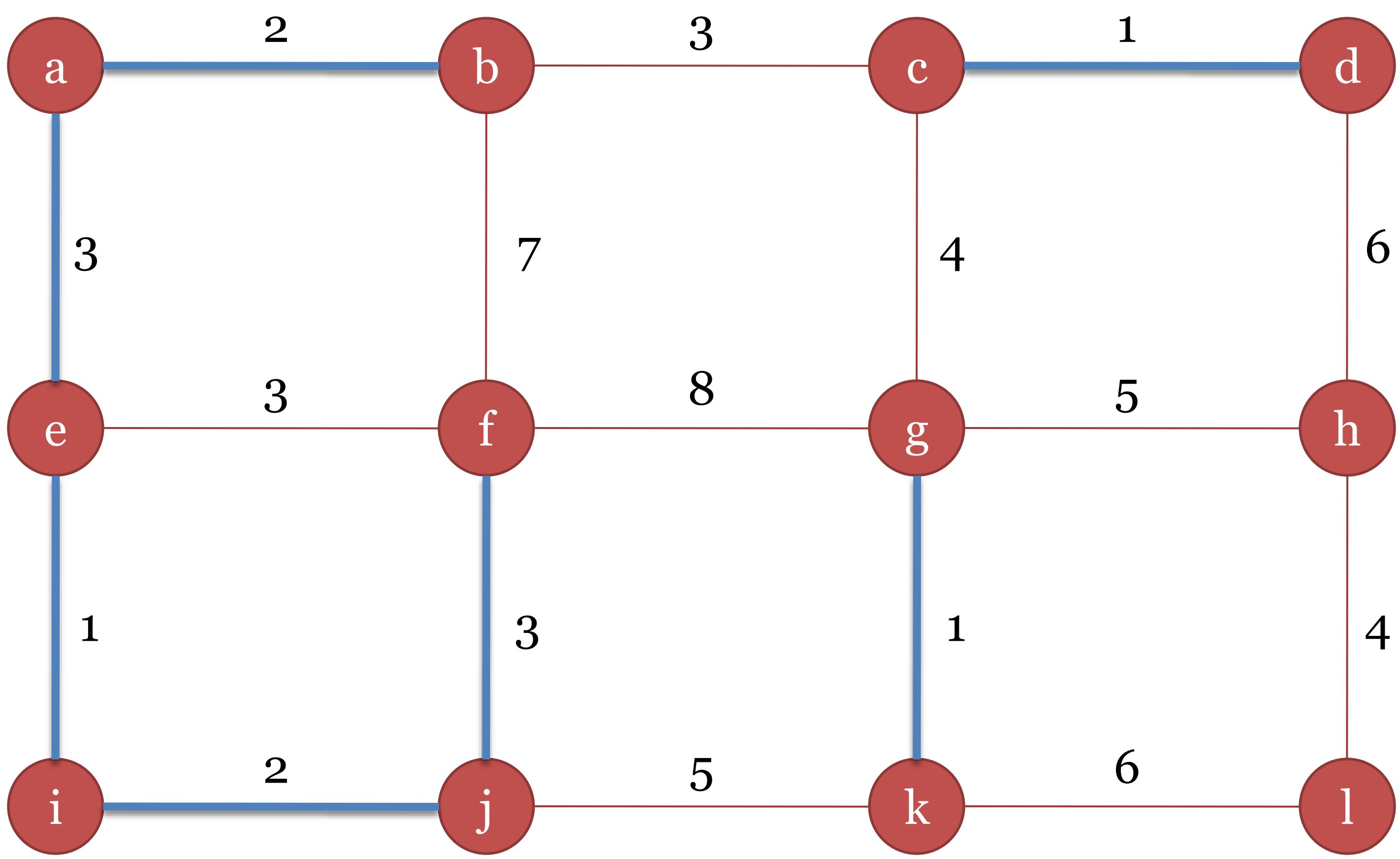
Tìm cây khung nhỏ nhất theo Kruskal



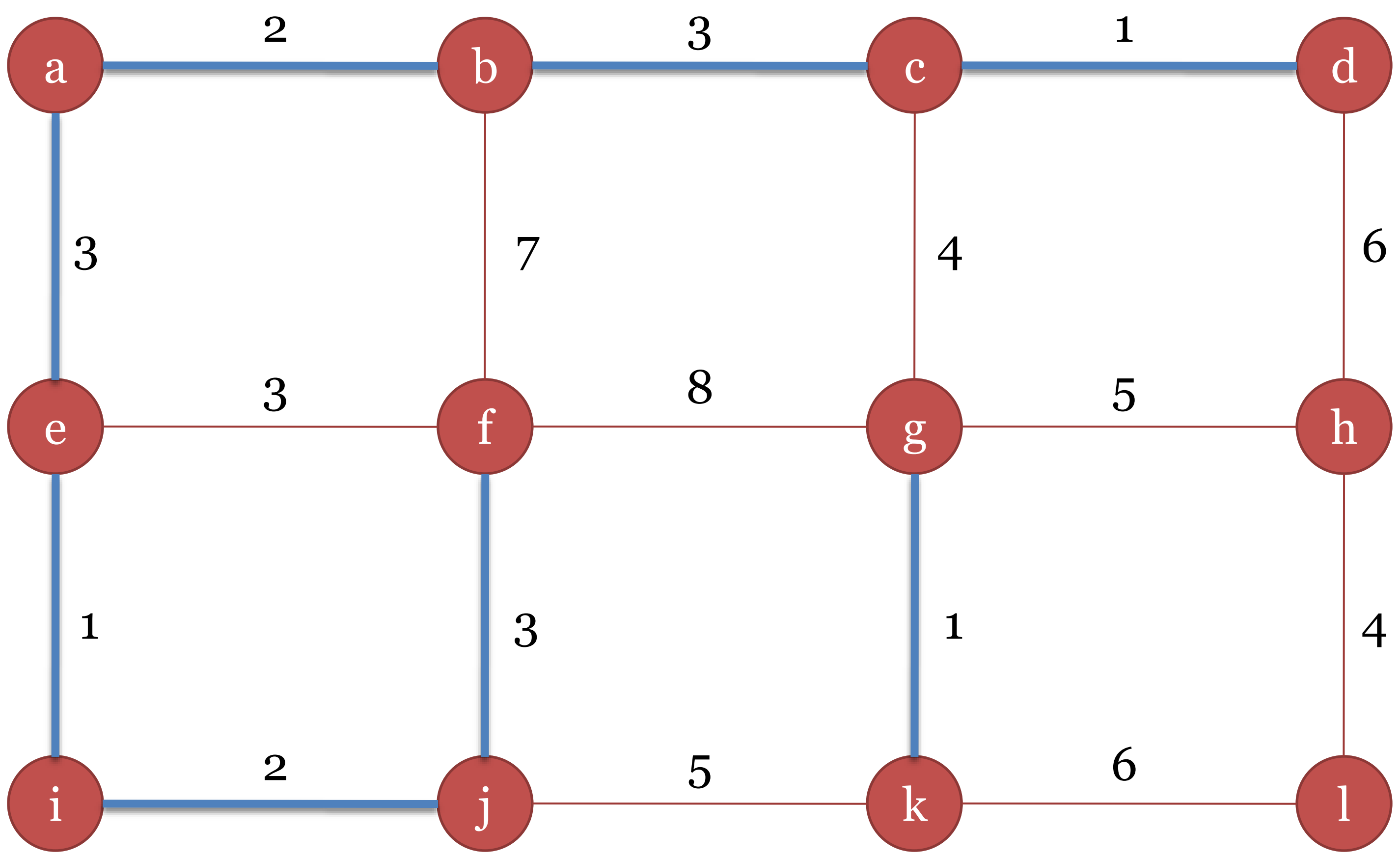
Tìm cây khung nhỏ nhất theo Kruskal



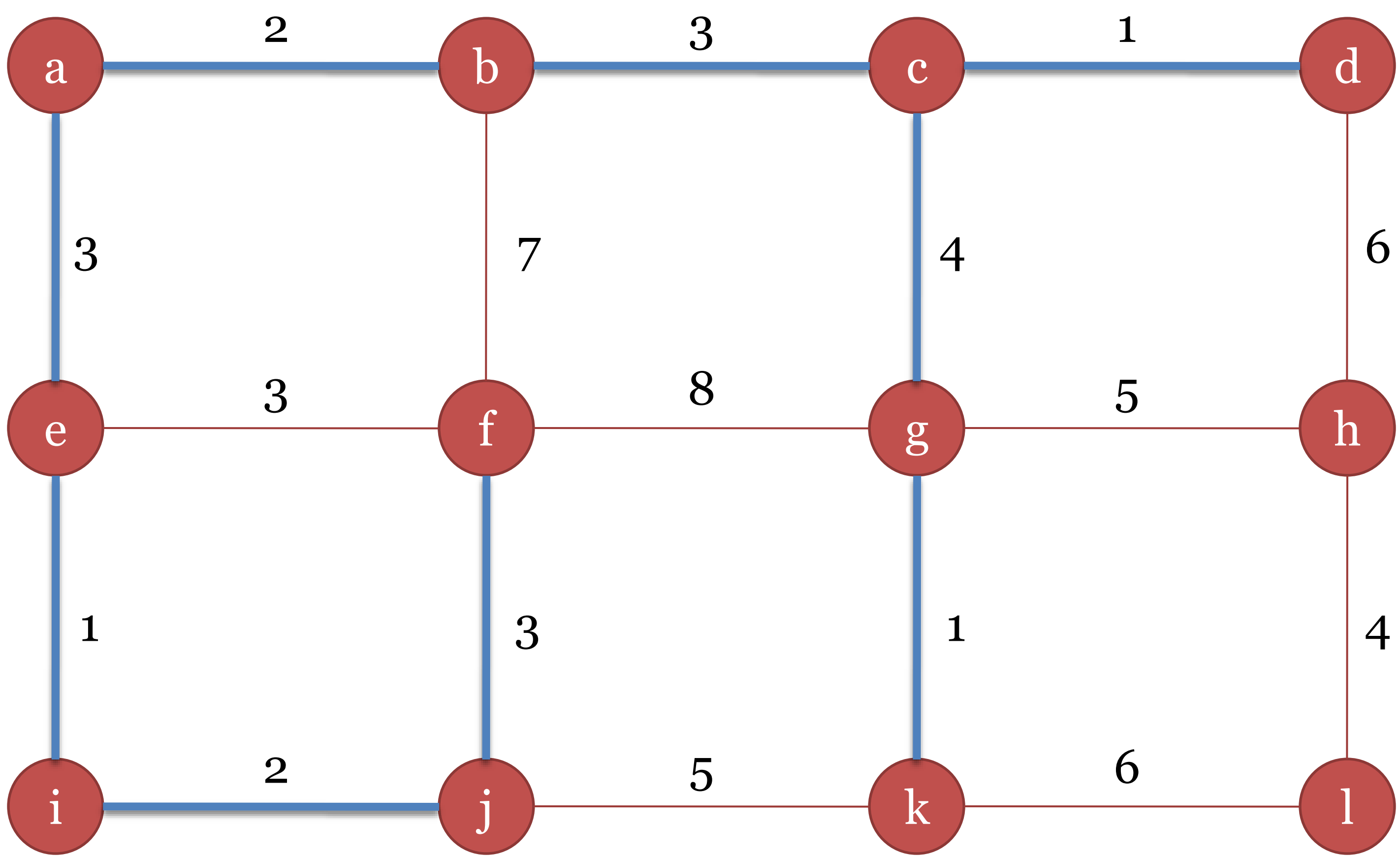
Tìm cây khung nhỏ nhất theo Kruskal



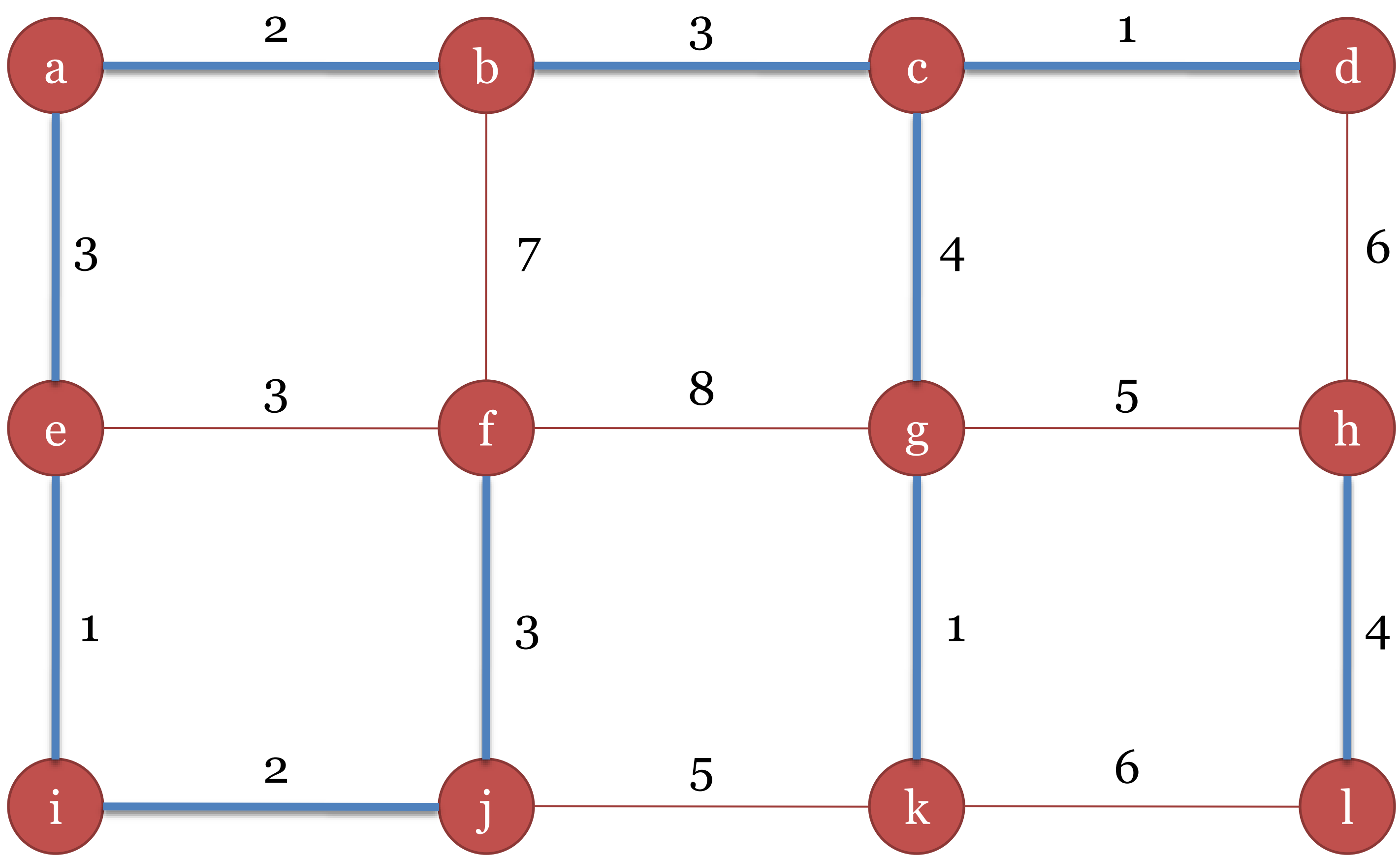
Tìm cây khung nhỏ nhất theo Kruskal



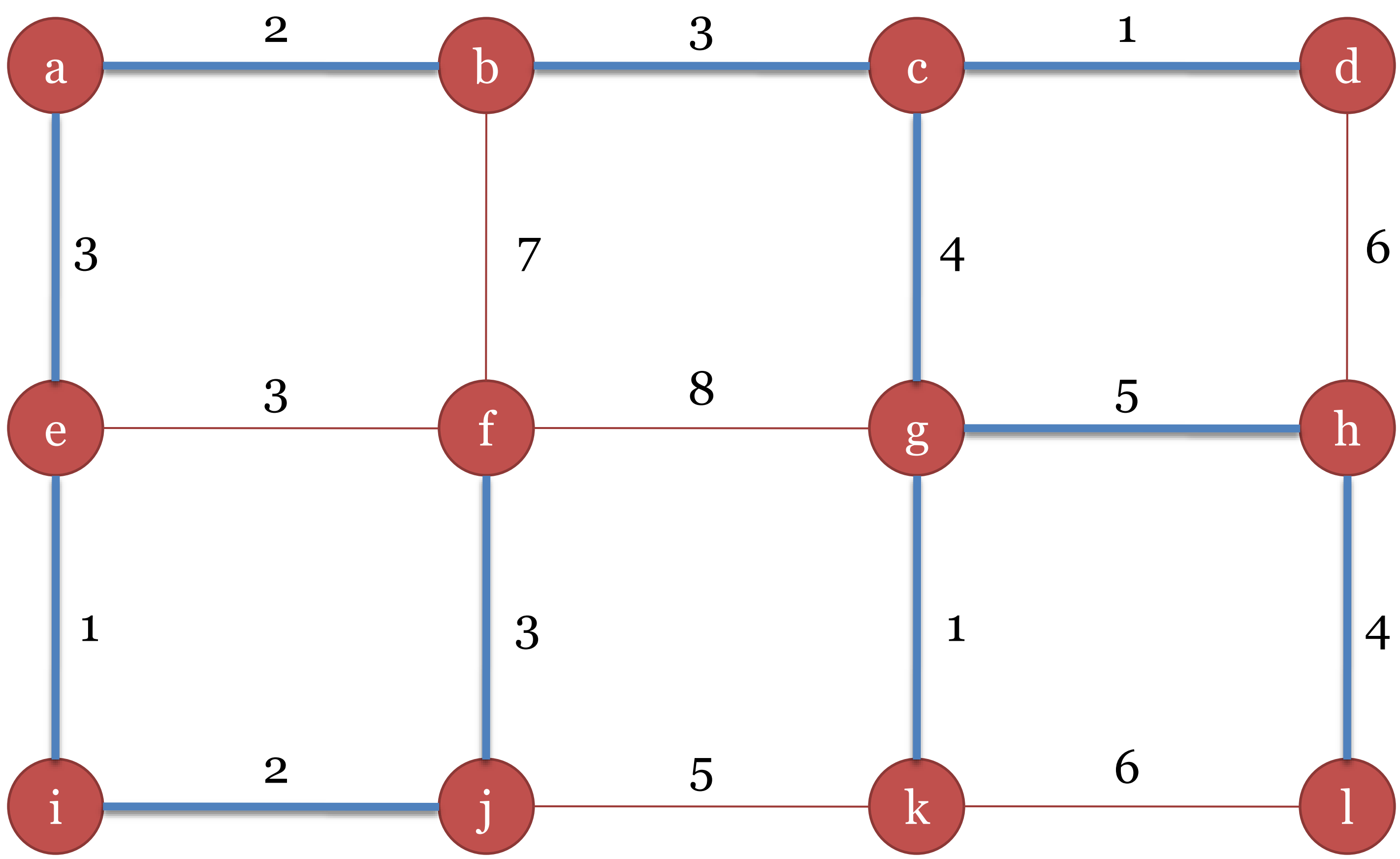
Tìm cây khung nhỏ nhất theo Kruskal



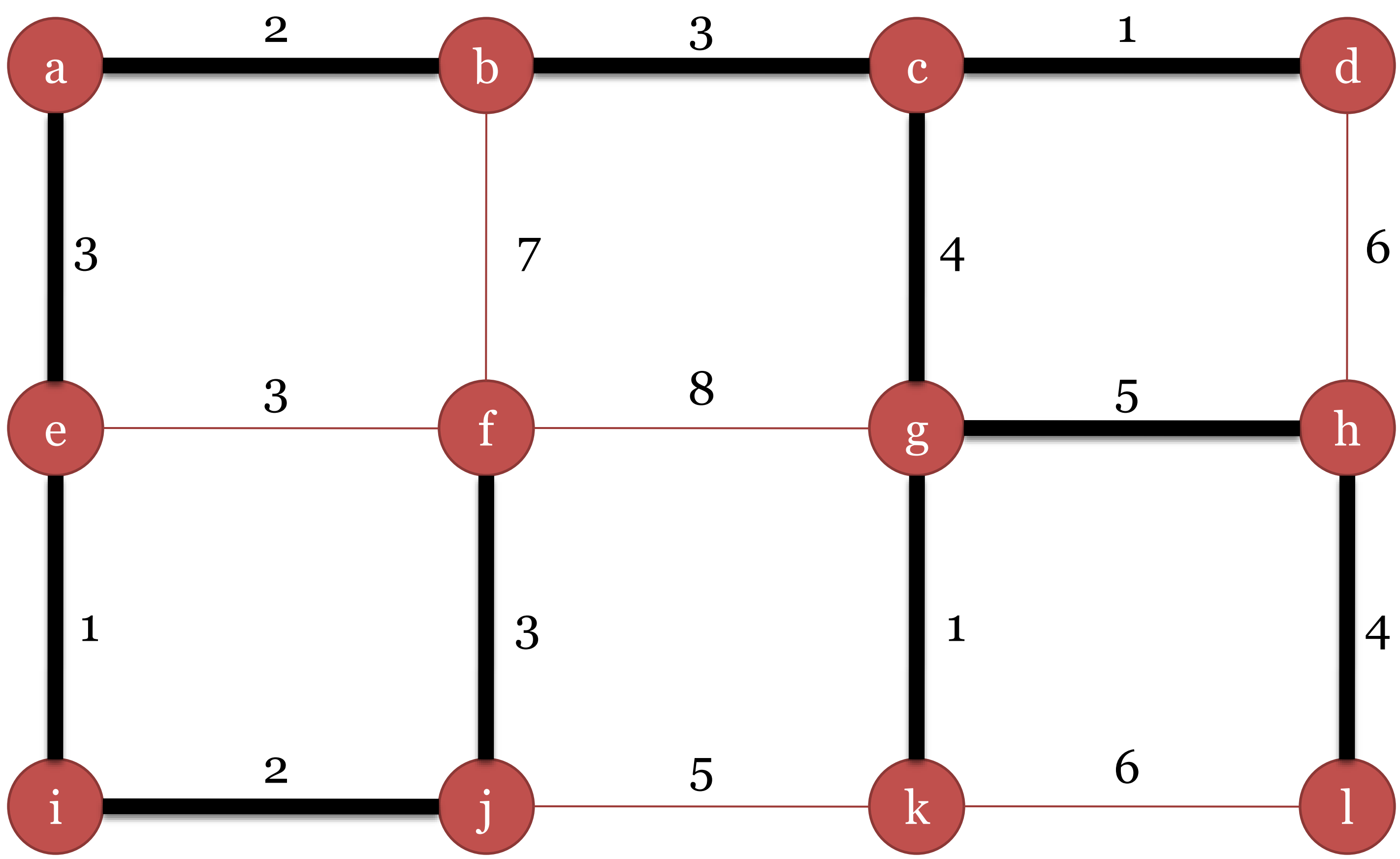
Tìm cây khung nhỏ nhất theo Kruskal



Tìm cây khung nhỏ nhất theo Kruskal

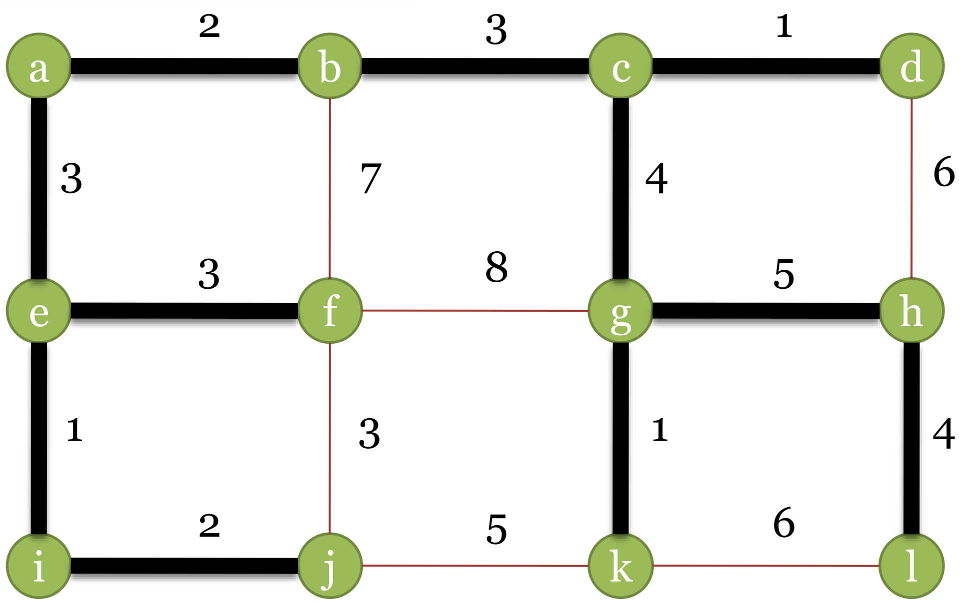


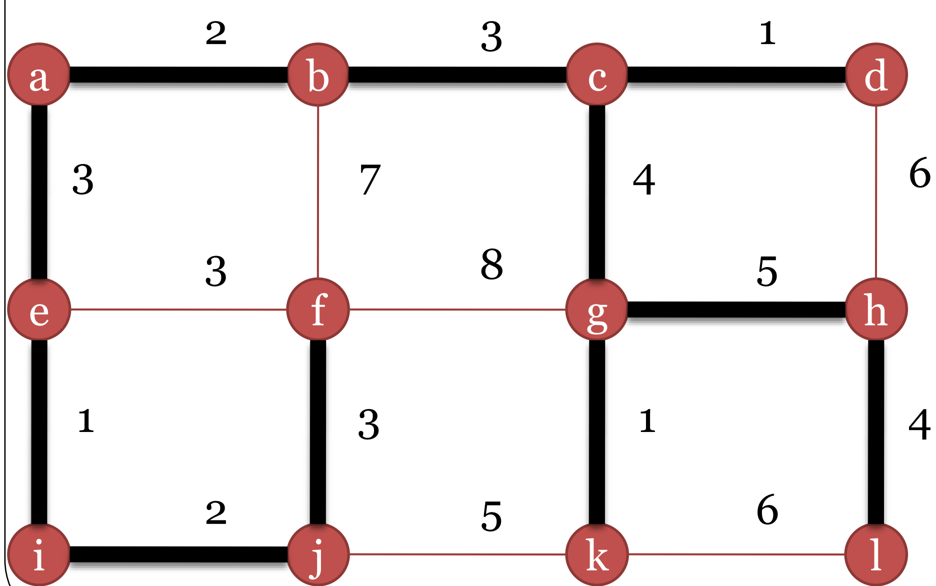
Tìm cây khung nhỏ nhất theo Kruskal



c(T) = 29

Kết quả Tìm cây khung nhỏ nhất



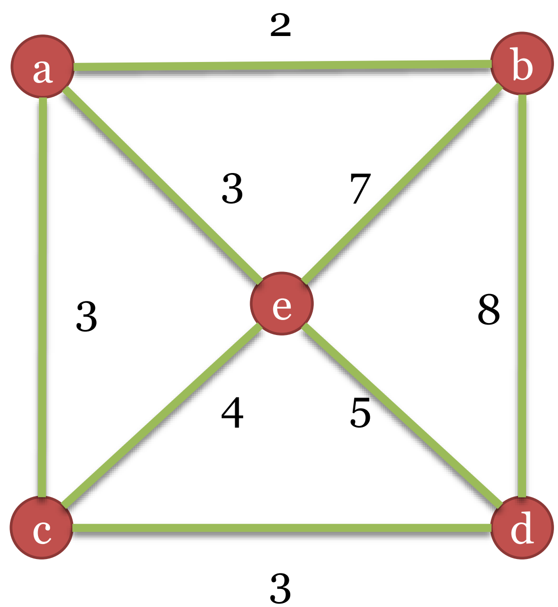


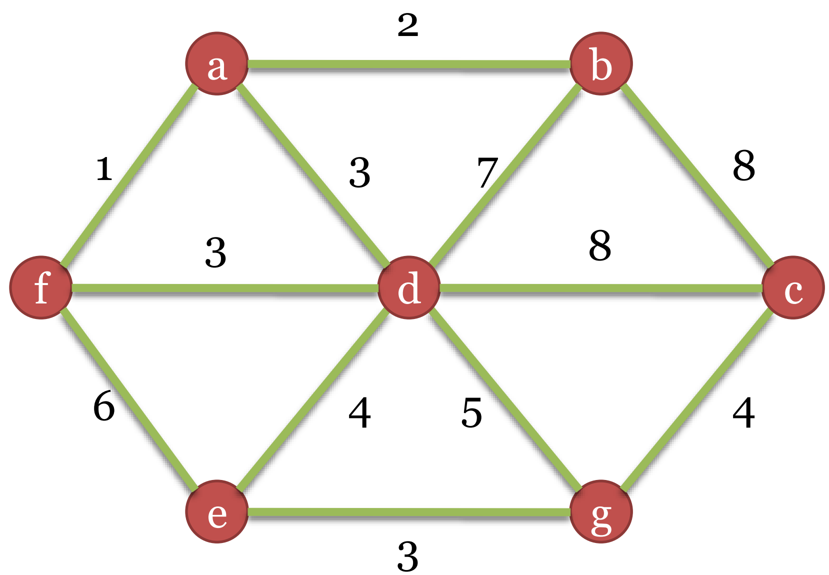
Bài tập

1. Dưới đây là một số giải thuật tìm cây khung của đồ thị liên thông G cho trước. Hãy cho biết các giải thuật này có đúng không?
   a) Bắt đầu với G. Nếu G có chu trình, hủy một cạnh thuộc chu trình này khỏi 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.

Bài tập

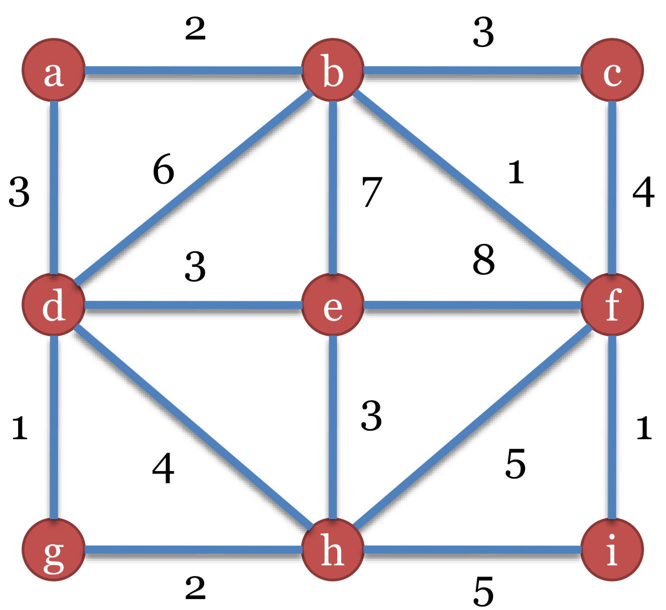
2. Sử dụng thuật toán Prim tìm cây khung nhỏ nhất cho các đồ thị có trọng số sau:

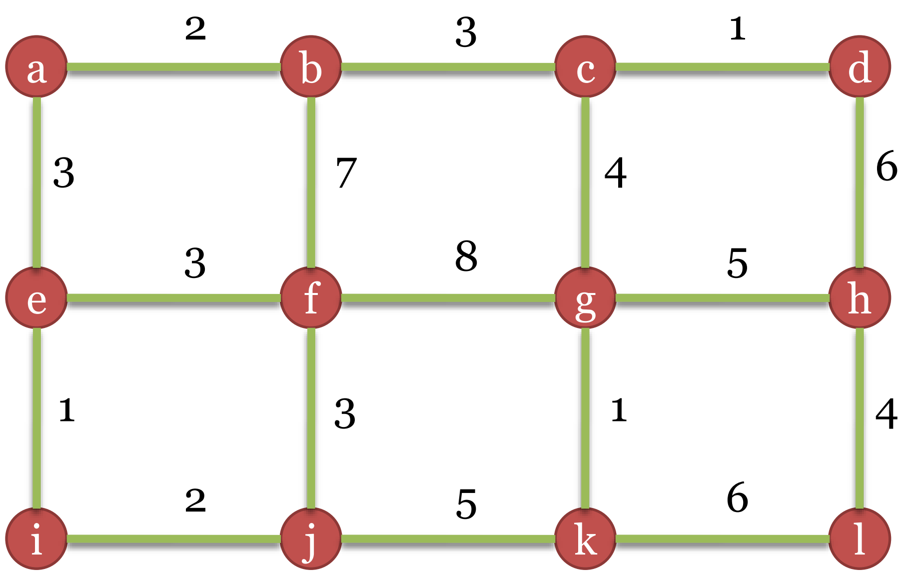




Bài tập

3. Sử dụng thuật toán Kruskal tìm cây khung nhỏ nhất cho các đồ thị sau:





Bài tập

4. Áp dụng giải thuật Prim và Kruskal tìm cây khung LỚN NHẤT cho đồ thị bên dưới. Minh họa kết quả cụ thể của từng bước thuật toán.

	1	2	3	4	5	6
1	0	33	17	∞	∞	∞
2	33	0	18	20	∞	∞
3	17	18	0	16	4	∞
4	∞	20	16	0	9	8
5	∞	∞	4	9	0	14
6	∞	∞	∞	8	14	0

Hỏi & đáp