

MÔN LÝ THUYẾT ĐỒ THỊ

Giảng viên: ThS. Mai Chiém Tuấn

CHƯƠNG V

CÂY VÀ CÂY KHUNG CỦA ĐỒ THỊ

Chương V: CÂY VÀ CÂY KHUNG CỦA ĐỒ THỊ



1

V.1 Các định nghĩa

2

V.2 Cây khung của đồ thị

3

V.3 Bài toán tìm cây khung ngắn nhất

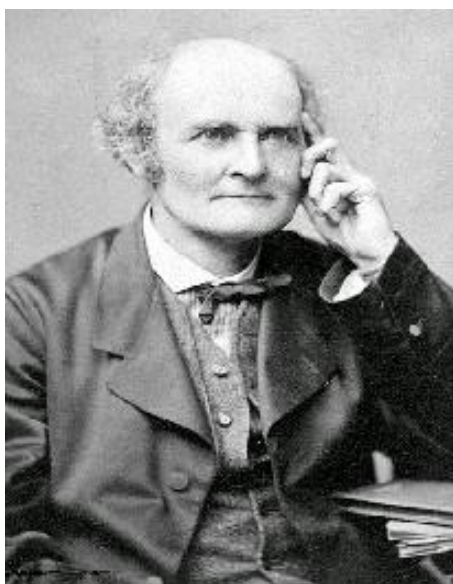
4

V.4 Bài tập

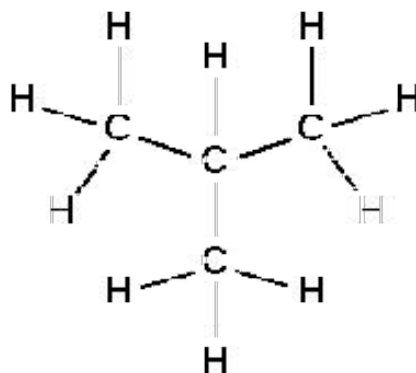
Giới thiệu



- ❖ Cây là một dạng của đồ thị được nhà toán học người Anh, Arthur Cayley, phát biểu và sử dụng từ năm 1857 cho việc xác định những cấu trúc hợp chất hóa học.



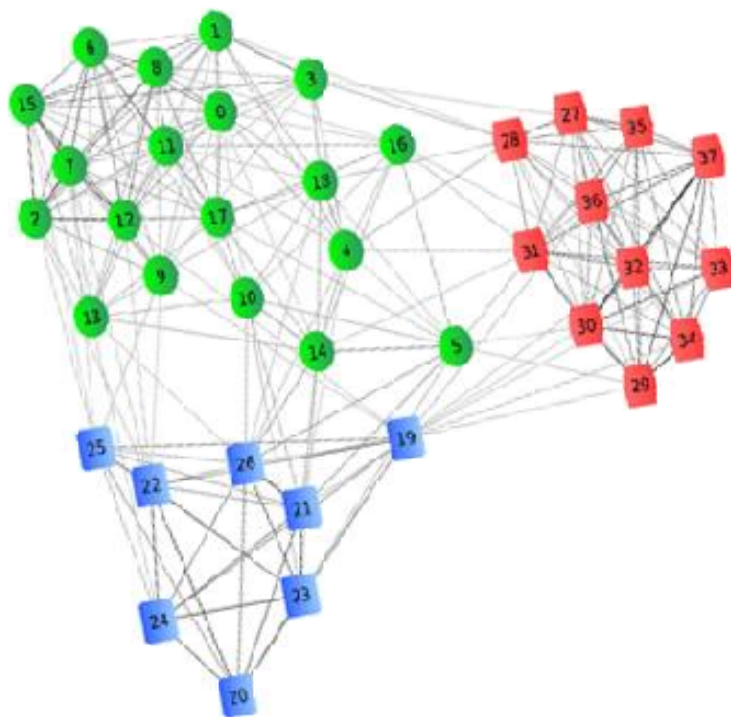
Arthur Cayley
(1821-1895)



Giới thiệu



- ❖ Sau đó cây được sử dụng nhiều trong khoa học máy tính để xây dựng các thuật toán hiệu quả; tính toán chi phí xây dựng mạng máy tính; mã hóa dữ liệu;

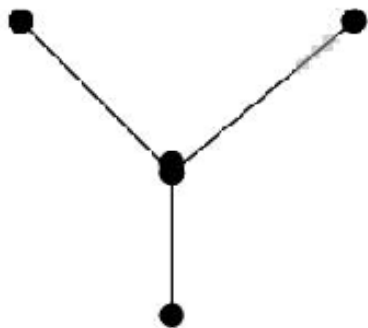


V.1 Các định nghĩa



Định nghĩa 5.1

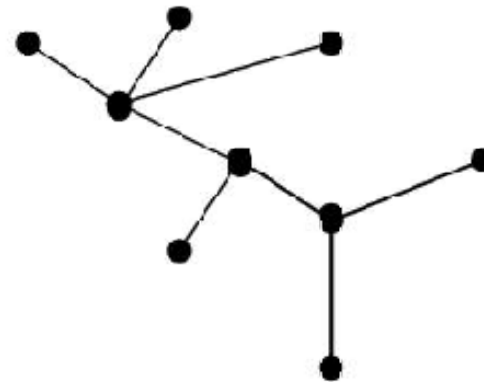
- ❖ Ta gọi **cây** là đồ thị vô hướng liên thông không có chu trình. Đồ thị không liên thông được gọi là **rừng**.
- ❖ Như vậy, **rừng** là đồ thị mà mỗi thành phần liên thông của nó là một **cây**.
- ❖ **Ví dụ 5.1:** Rừng gồm 3 cây $T1, T2, T3$



T1



T2



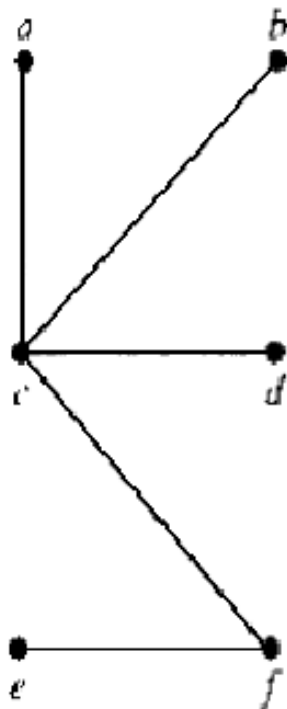
T3

V.1 Các định nghĩa

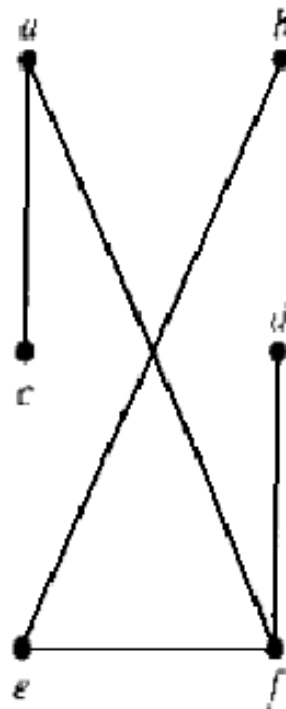


Ví dụ 5.2

❖ Trong các đồ thị sau, đồ thị nào là cây?



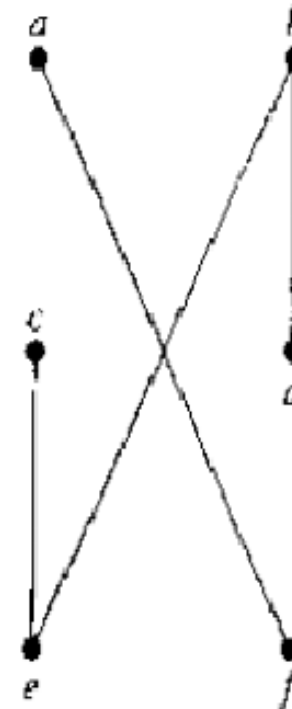
G_1



G_2



G_3



G_4

- G_1, G_2 là cây.
- G_3 không là cây do có chứa chu trình, G_4 không liên thông

V.1 Các định nghĩa



Định lý 5.1

- ❖ Giả sử $G=(V,E)$ là đồ thị vô hướng n đỉnh. Khi đó các mệnh đề sau đây là tương đương:
- T là cây;
 - T không chứa chu trình và có $n-1$ cạnh;
 - T liên thông và có $n-1$ cạnh;
 - T liên thông và mỗi cạnh của nó đều là cầu;
 - Hai đỉnh bất kỳ của T được nối với nhau bởi đúng một đường đi đơn;
 - T không chứa chu trình nhưng nếu thêm vào một cạnh ta thu được đúng một chu trình, ...

V.2 Cây khung của đồ thị



Định nghĩa 5.2

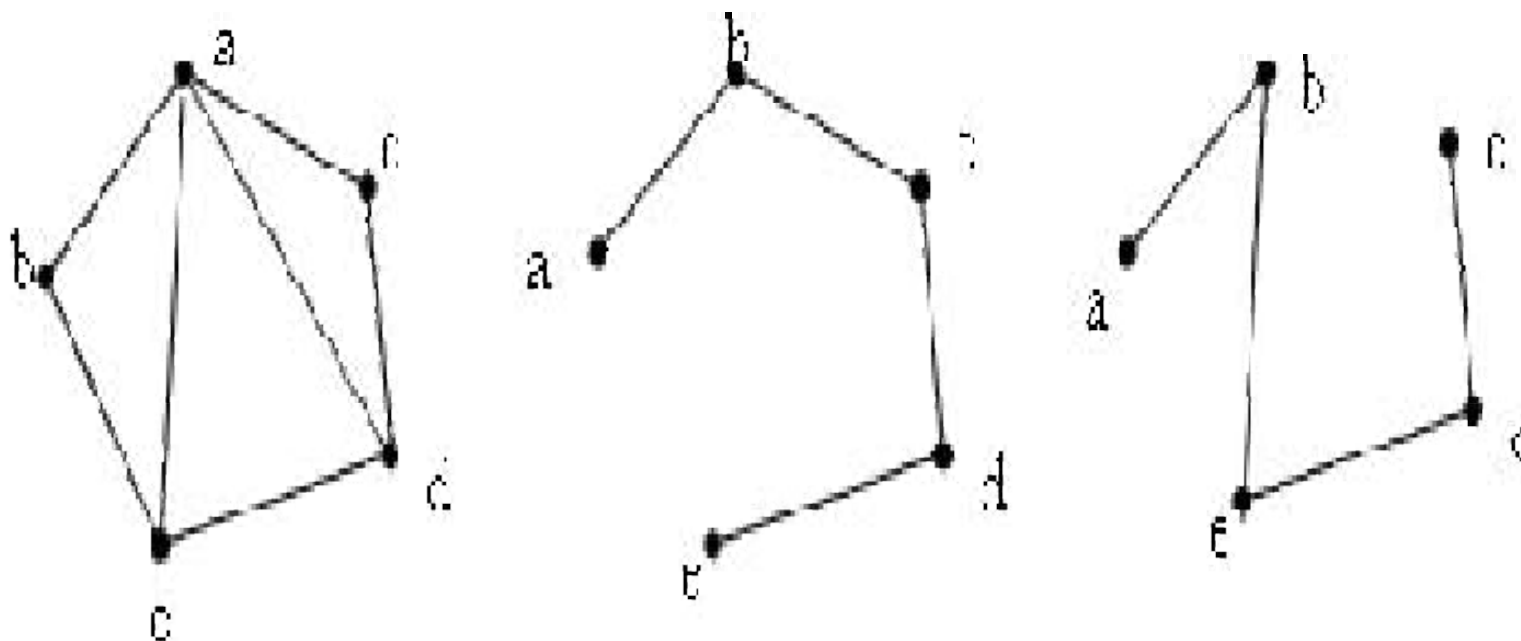
- ❖ Cho G là đồ thị vô hướng liên thông. Ta gọi đồ thị con T của G là một **cây khung** của G (*Cây bao trùm*) nếu T thoả mãn hai điều kiện:
 - T là một cây;
 - Tập đỉnh của T bằng tập đỉnh của G .
- ❖ Hai bài toán cơ bản về cây:
 - Cho đồ thị vô hướng $G = \langle V, E \rangle$. Xây dựng một cây khung của đồ thị bắt đầu tại đỉnh $u \in V$. **Được giải quyết bằng thuật toán DFS hoặc BFS.**
 - Cho đồ thị vô hướng $G = \langle V, E \rangle$ có trọng số. Hãy xây dựng cây khung có độ dài nhỏ nhất. **Được giải quyết bằng thuật toán Kruskal hoặc PRIM.**

V.2 Cây khung của đồ thị



Ví dụ 5.3

❖ Đồ thị và các cây khung của nó.



V.2 Cây khung của đồ thị



Cách tạo cây khung của đồ thị

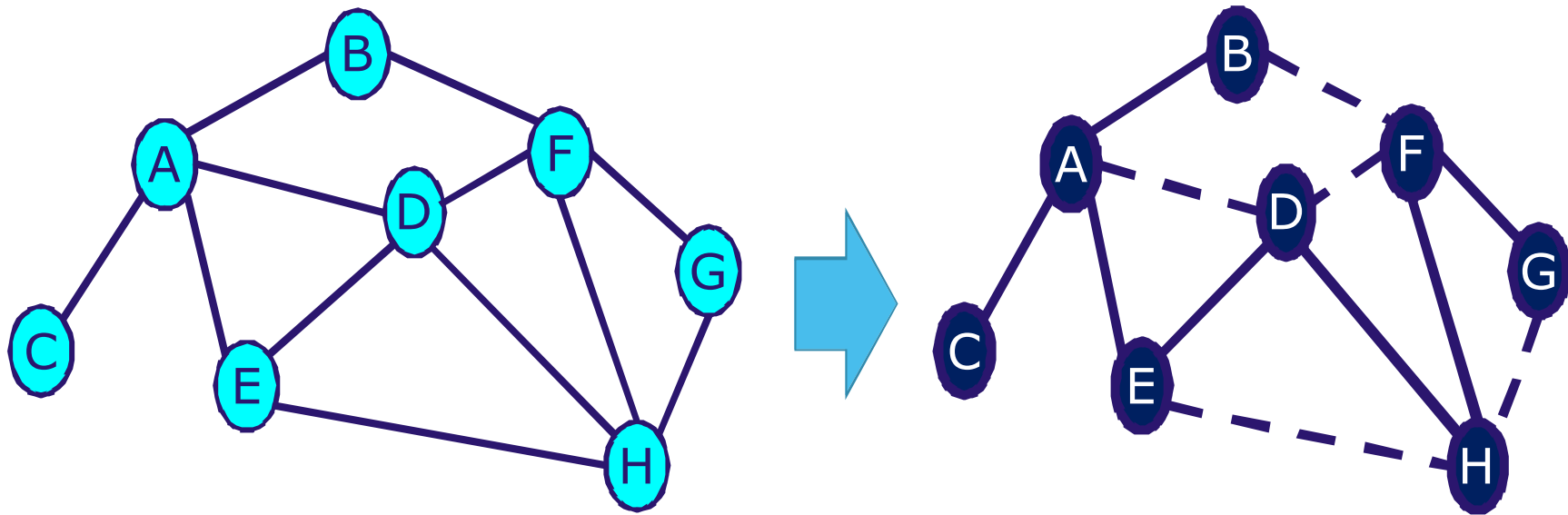
- ❖ Trong đồ thị liên thông G , chúng ta thực hiện loại bỏ một cạnh nằm trên một chu trình nào đó sẽ tạo ra đồ thị G' vẫn có tính liên thông.
- ❖ Thực hiện tiếp việc loại bỏ các cạnh ở các chu trình khác cho đến khi đồ thị T không còn chu trình nhưng vẫn liên thông thì chúng ta thu được một **cây nối tất cả các đỉnh** của G - gọi là **cây khung của đồ thị**.

V.2 Cây khung của đồ thị



Ví dụ 5.4

- ✦ Cho đồ thị G trong hình dưới đây, hãy thực hiện tìm các cây khung của đồ thị G .

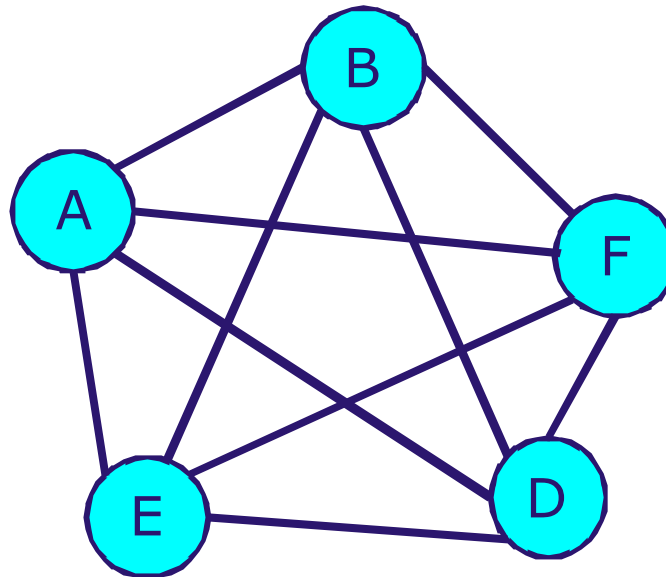


V.2 Cây khung của đồ thị



Định lý 5.2 (Cayley)

- ❖ Số lượng cây khung của đồ thị K_n là n^{n-2} .
- ❖ **Ví dụ 5.5:** Một đồ thị đầy đủ có 5 đỉnh sẽ có số cây khung là $5^3 = 125$.



Vậy làm thế nào để tìm được cây khung có độ dài ngắn nhất cho đồ thị có trọng số ?

V.3 Tìm cây khung ngắn nhất



Bài toán tìm cây khung ngắn nhất

- ❖ Cho $G=(V,E)$ là đồ thị vô hướng, liên thông có trọng số. Độ dài $m(T)$ của cây khung T là tổng trọng số các cạnh của cây:

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

- ❖ **Bài toán:**

- Trong số tất cả các cây khung của đồ thị G , hãy tìm ra cây khung có độ dài ngắn nhất - gọi là **cây khung ngắn nhất của đồ thị**.

V.3 Tìm cây khung ngắn nhất



Các bài toán thực tế

❖ Bài toán nối mạng máy tính:

- Với mạng máy tính gồm n máy đánh số từ 1 đến n . Biết chi phí nối máy i với máy j là $m(i,j)$ (chi phí phụ thuộc vào độ dài cáp nối cần sử dụng). Hãy tìm cách nối mạng sao cho tổng chi phí là nhỏ nhất.

❖ Bài toán xây dựng hệ thống đường sắt:

- Chúng ta muốn xây dựng một hệ thống đường sắt nối n thành phố để hành khách từ một thành phố có thể đi đến bất kỳ các thành phố còn lại. Yêu cầu thiết kế để chi phí xây dựng hệ thống đường đi là nhỏ nhất.

V.3 Tìm cây khung ngắn nhất



V.3.1 Thuật toán **Prim** (năm 1957)

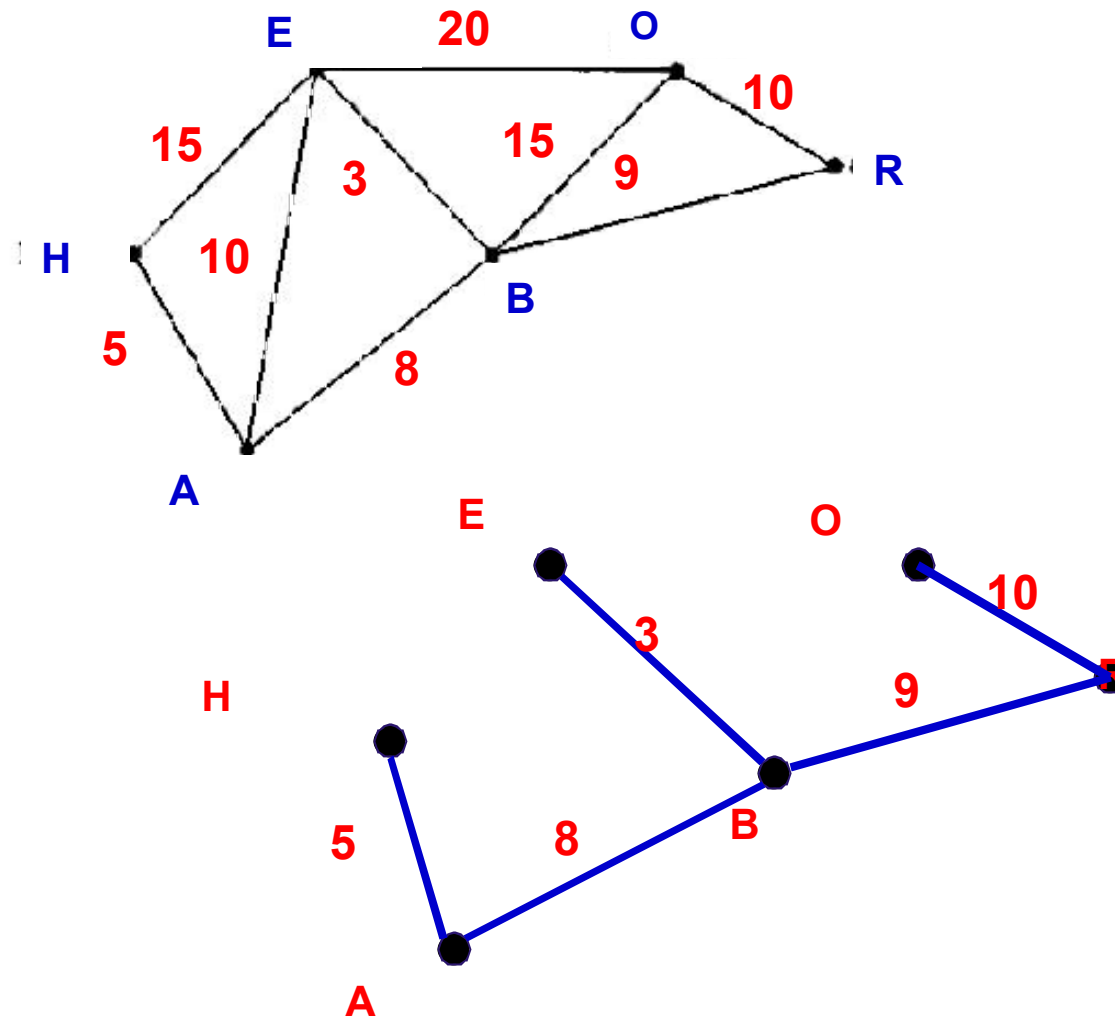
❖ *Ý tưởng:*

- Xuất phát từ 1 đỉnh bất kỳ. Đưa đỉnh này vào cây khung T.
- Tại mỗi bước, luôn chọn cạnh có trọng số nhỏ nhất trong số các cạnh liên thuộc với một đỉnh trong T (đỉnh còn lại nằm ngoài T).
- Đưa cạnh mới chọn và đỉnh đầu của nó vào cây T.
- Lặp lại quá trình trên cho đến khi đưa đủ $n-1$ cạnh vào T.

V.3 Tìm cây khung ngắn nhất

V.3.1 Thuật toán Prim (năm 1957)

Ví dụ 5.6



V.3 Tìm cây khung ngắn nhất

V.3.1 Thuật toán Prim (năm 1957)



(* Khởi tạo *)

Chọn s là một đỉnh nào đó của đồ thị

$V_H := \{s\};$ (* Tập những đỉnh đã đưa vào cây *)

$T := \emptyset;$ (* Tập cạnh của cây *)

$d[s] = 0;$ $near[s] = s;$

For $v \in V \setminus V_H$ do

 Begin

$d[v] := a[s,v];$

$near[v] := s;$

 End;

(* Bước lặp *)

Stop := False;

While (not Stop) do

 Begin

 Tìm $u \in V \setminus V_H$ thỏa mãn $d[u] = \min\{d[v] : v \in V \setminus V_H\};$

$V_H := V_H \cup \{u\};$

$T := T \cup \{(u, near[u])\};$

 If $|V_H| = n$ then

 Begin

$H := (V_H, T)$ là cây khung của đồ thị.

 Stop := True;

 End;

 Else

 For $v \in V \setminus V_H$ do

 If $d[v] > a[u,v]$ then

 Begin

$d[v] := a[u,v];$

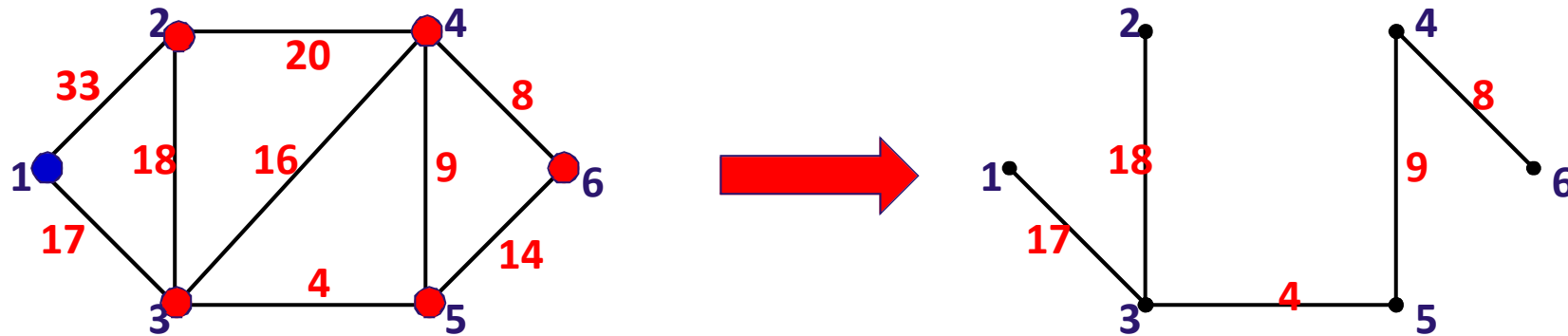
$near[v] := u;$

 End;

 End;

V.3 Tìm cây khung ngắn nhất

V.3.1 Thuật toán Prim (năm 1957)



Bước lặp	Đỉnh 1	Đỉnh 2	Đỉnh 3	Đỉnh 4	Đỉnh 5	Đỉnh 6	V_H	T
Khởi tạo	[0,1]	[33,1]	[17,1]*	$[\infty,1]$	$[\infty,1]$	$[\infty,1]$	1	$\emptyset$
1	-	[18,3]	-	[16,3]	[4,3]*	$[\infty,1]$	1,3	(3,1)
2	-	[18,3]	-	[9,5]*	-	[14,5]	1,3,5	(3,1),(5,3)
3	-	[18,3]	-	-	-	[8,4]*	1,3,5,4	(3,1),(5,3),(4,5)
4	-	[18,3]*	-	-	-	-	1,2,3,4,6	(3,1),(5,3),(4,5) (6,4)
5	-	-	-	-	-	-	1,2,3,4,6,2	(3,1),(5,3),(4,5) (6,4),(2,3)

V.3 Tìm cây khung ngắn nhất



V.3.2 Thuật toán **Kruskal**

❖ *Ý tưởng:*

- Lần lượt xét các cạnh theo thứ tự trọng số tăng dần
- Ứng với mỗi cạnh đang xét, ta thử đưa nó vào cây khung T:
 - Nếu không tạo thành chu trình với các cạnh đã chọn thì chấp nhận cạnh mới này và đưa vào cây.
 - Nếu tạo thành chu trình với các cạnh đã chọn thì bỏ qua và xét cạnh kế tiếp.
- Cứ tiếp tục như vậy cho đến khi tìm đủ $n-1$ cạnh để đưa vào cây T.

V.3 Tìm cây khung ngắn nhất

V.3.2 Thuật toán Kruskal

Ví dụ 5.8



Trọng số	Cạnh
3	(E,B)
5	(A,H)
8	(A,B)
9	(B,R)
10	(A,E)
10	(R,O)
15	(E,H)
15	(B,O)
20	(E,O)

← Chọn

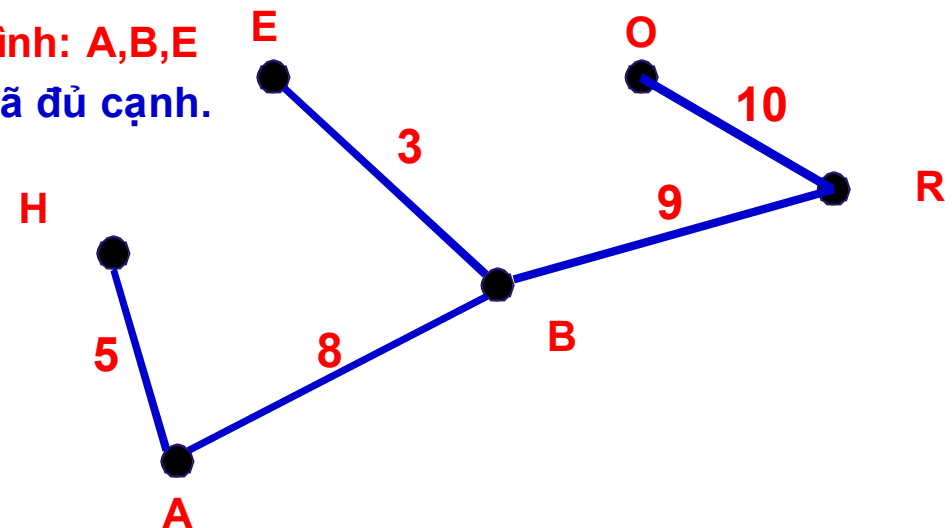
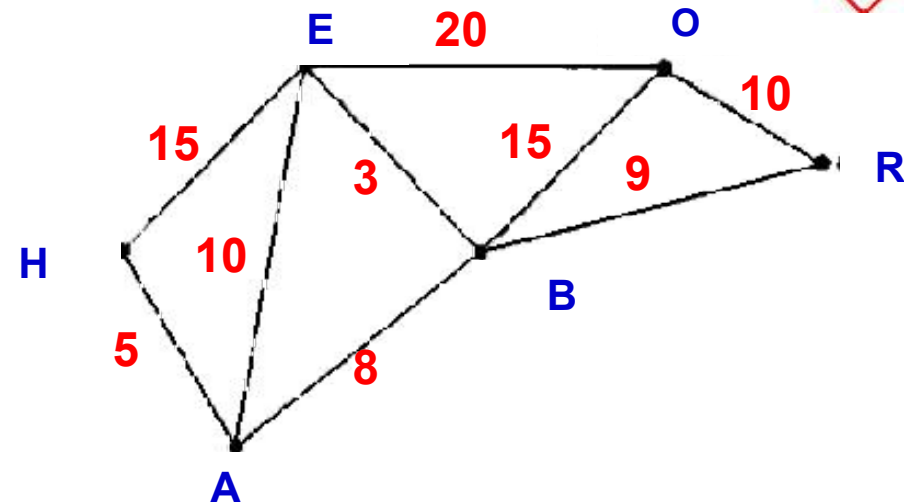
← Chọn

← Chọn

← Chọn

← Không chọn vì tạo chu trình: A,B,E

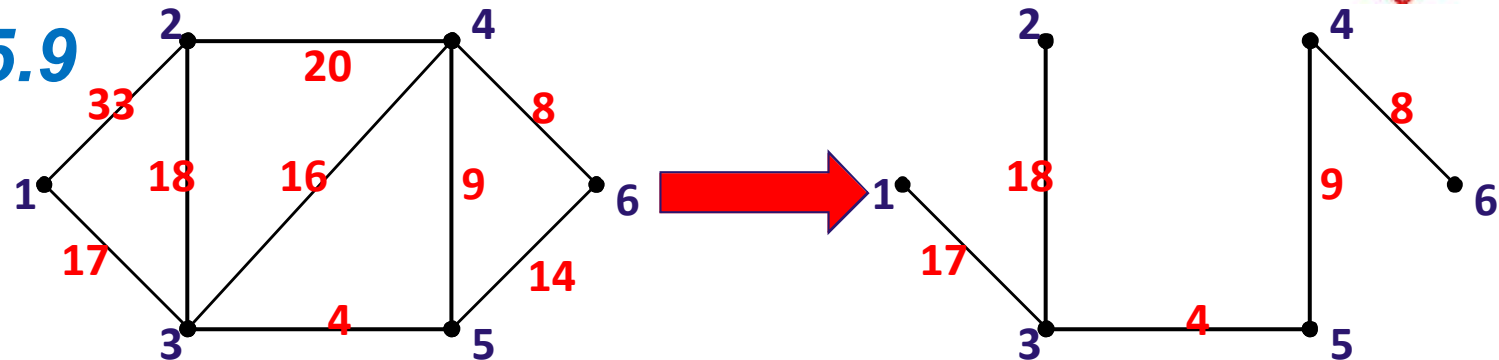
← Chọn, Dừng vì đã đủ cạnh.



V.3 Tìm cây khung ngắn nhất

V.3.2 Thuật toán Kruskal

Ví dụ 5.9



Trọng số	Cạnh	
4	(3,5)	← Chọn
8	(4,6)	← Chọn
9	(4,5)	← Chọn
14	(5,6)	← Không chọn vì tạo chu trình: 4 5 6 4
16	(3,4)	← Không chọn vì tạo chu trình: 3 4 5 3
17	(1,3)	← Chọn
18	(2,3)	← Chọn. Dừng vì đã đủ cạnh.
20	(2,4)	
33	(1,2)	

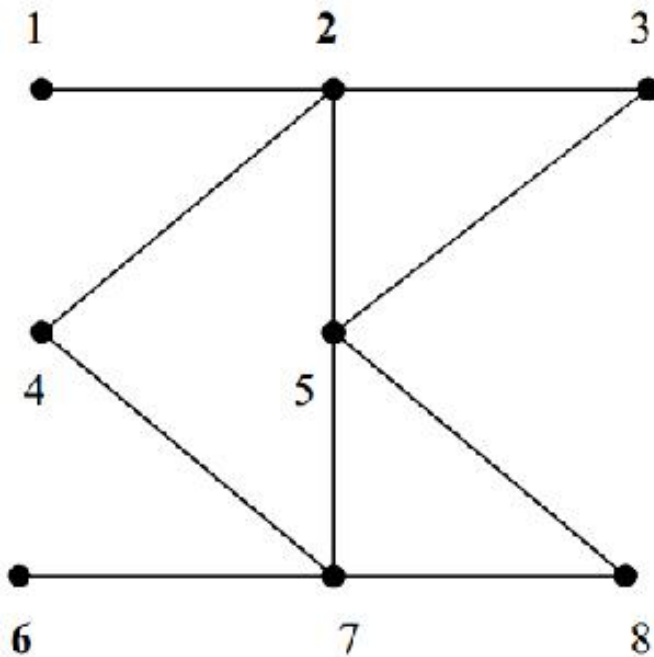


V.4 Bài tập

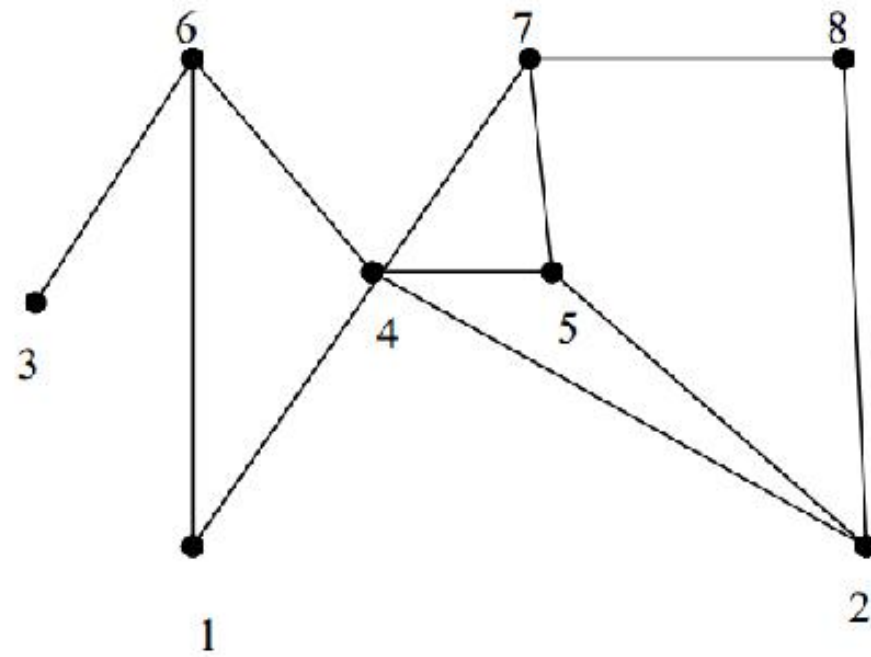


Bài 1. Giả sử đồ thị G liên thông, có 13 đỉnh và 20 cạnh. Hỏi cây khung của G có bao nhiêu đỉnh? có bao nhiêu cạnh?.

Bài 2. Tìm cây khung của đồ thị như hình sau theo phương pháp DFS, BFS (chọn đỉnh 1 làm gốc).



Hình 1



Hình 2

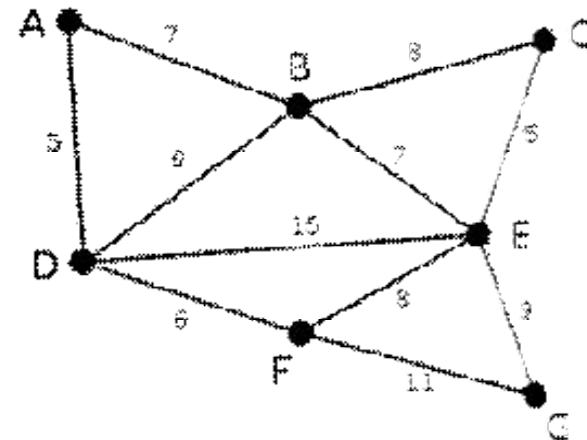
V.4 Bài tập



Bài 3. Tìm cây khung nhỏ nhất bằng thuật toán **Kruskal**, **Prim** của đồ thị gồm các đỉnh A, B, C, D, E, F, H, I được cho bởi ma trận trọng số sau.

	A	B	C	D	E	F	H	I
A	∞	15	16	19	23	20	32	18
B	15	∞	33	13	34	19	20	12
C	16	33	∞	13	29	21	20	19
D	19	13	13	∞	22	30	21	11
E	23	34	29	22	∞	34	23	21
F	20	19	21	30	34	∞	17	18
H	32	20	20	21	23	17	∞	14
I	18	12	19	11	21	18	14	∞

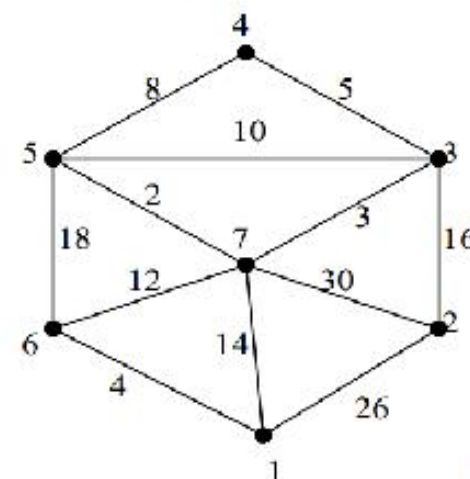
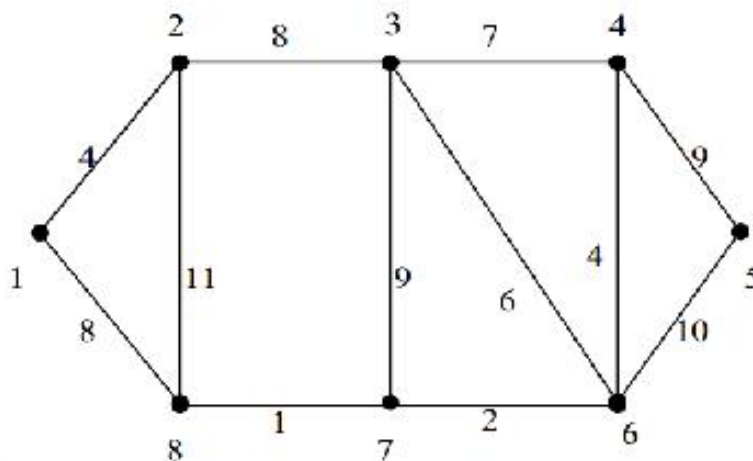
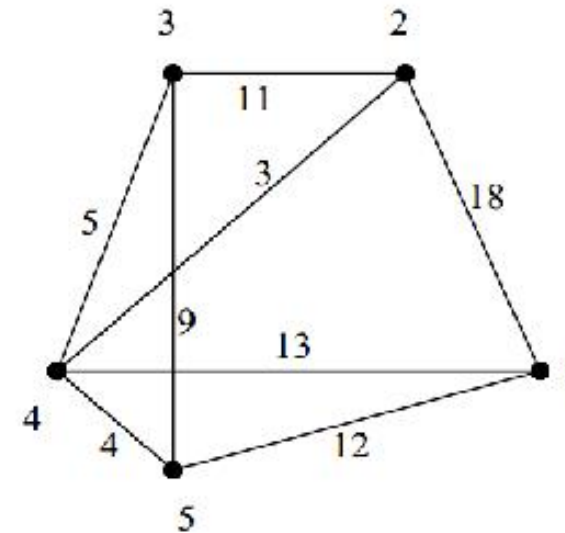
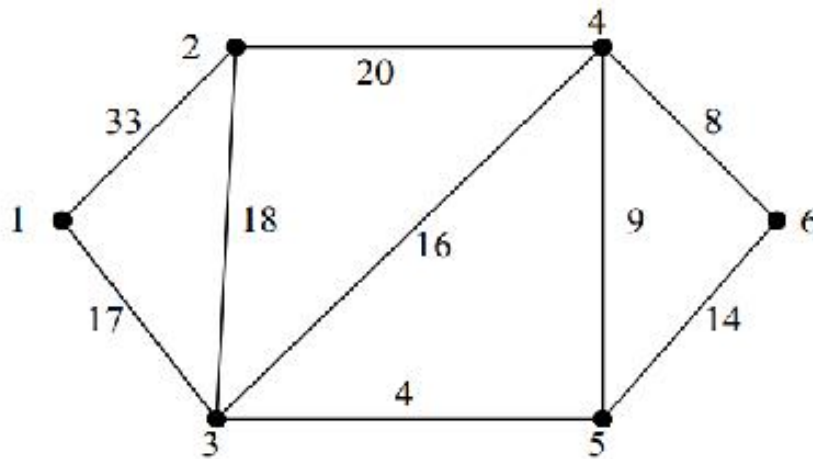
Bài 4. Tìm cây khung nhỏ nhất bằng thuật toán **Kruskal**, **Prim** của đồ thị gồm các đỉnh A, B, C, D, E, F, H, I được cho bởi đồ thị sau.



V.4 Bài tập



Bài 5. Tìm cây khung nhỏ nhất bằng thuật toán **Kruskal**,
Prim của các đồ thị sau:



Q & A?