

Chương 4

CÂY (TREE)

TS.Nguyễn Thị Phương Trâm

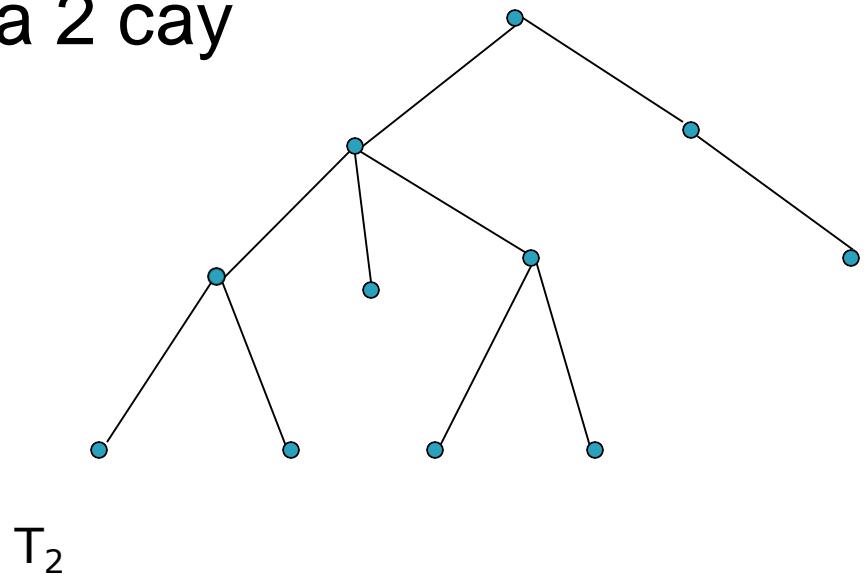
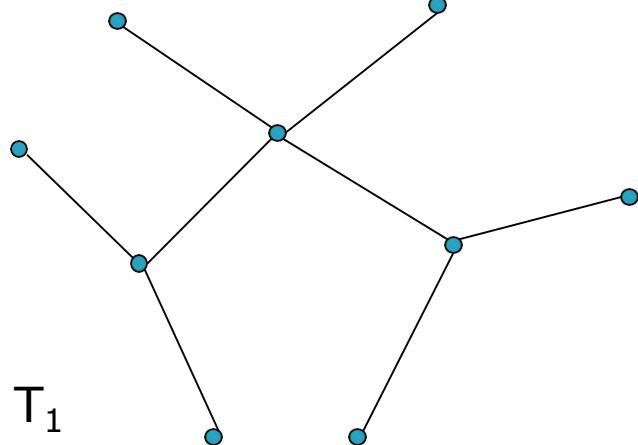
Tài liệu tham khảo:

- Silde bài giảng ThS. Trần Quốc Việt
- Silde bài giảng ThS. Lê Phi Hùng
- Nguyễn Cam, Chu Đức Khánh, *Lý thuyết Đồ thị*, 1998.
- Kenneth H. Rosen, *Discrete Mathematics and Its Applications*

1. Định nghĩa và các tính chất cơ bản

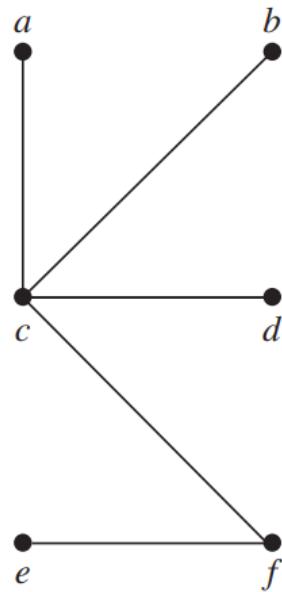
□ Định nghĩa: Cây (Tree), còn gọi là cây tự do (free tree) là một đồ thị vô hướng liên thông và không có chu trình

Ví dụ: T_1 và T_2 sau đây là 2 cây

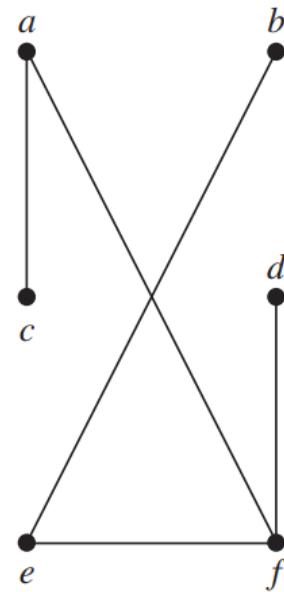


Định nghĩa và các tính chất cơ bản

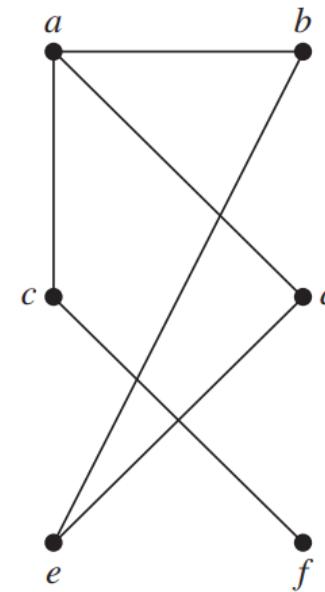
□ Định lý 1: Giữa 2 đỉnh bất kỳ trong cây T luôn có một và chỉ một đường đi trong T nối chúng



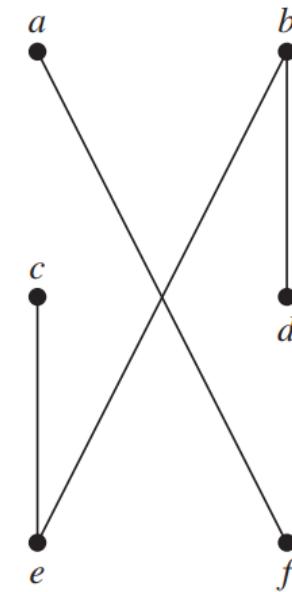
G_1



G_2



G_3

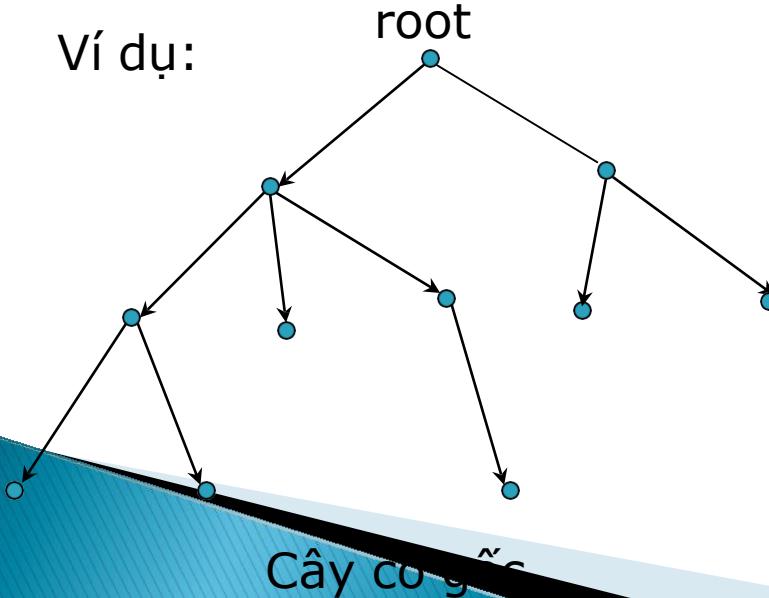


G_4

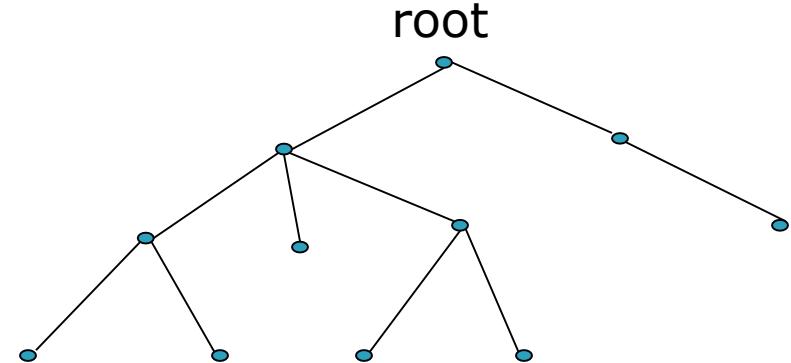
Định nghĩa và các tính chất cơ bản (tt)

- ❖ **Định nghĩa:** Cây có gốc (**rooted tree**) là một cây có hướng, trên đó đã chọn một đỉnh là gốc (root) của cây và các cạnh định hướng sao cho với mọi đỉnh luôn có một đường đi từ gốc đến đỉnh đó

Ví dụ:



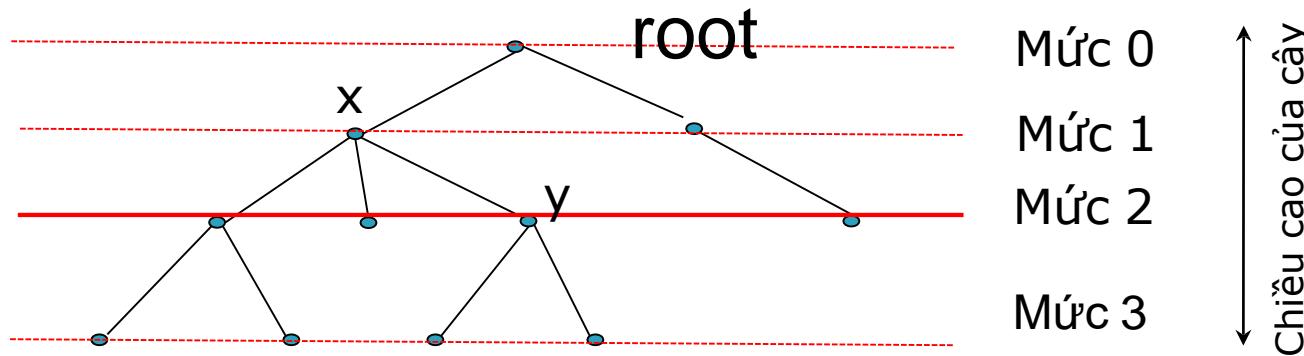
*Một cây tự do có thể chọn
một đỉnh bất kỳ làm gốc để
trở thành cây có gốc*



Định nghĩa và các tính chất cơ bản (tt)

Xét cây có gốc T

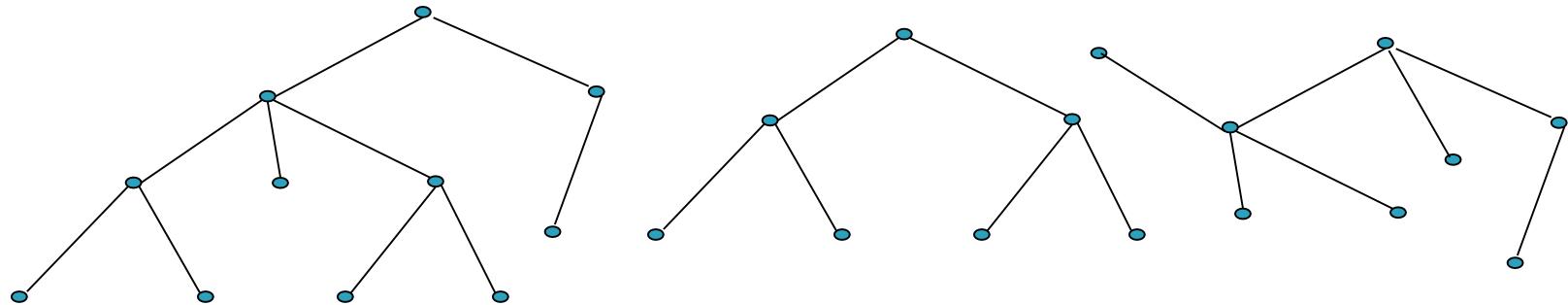
- Mức của đỉnh: Khoảng cách từ gốc đến nút đó
- Chiều cao của cây: Mức lớn nhất của một đỉnh bất kỳ



- Nếu (xy) là cạnh của T: ta gọi x đỉnh cha (parent) của y, y là đỉnh con (child) của x
- Lá (leaves): Những đỉnh **không có cây con**.
- Đỉnh trong (internal vertices): là những đỉnh có ít nhất 1 cây con

Định nghĩa và các tính chất cơ bản (tt)

Định nghĩa: Tập hợp các cây đôi một không có đỉnh chung gọi là một rừng (Forest)



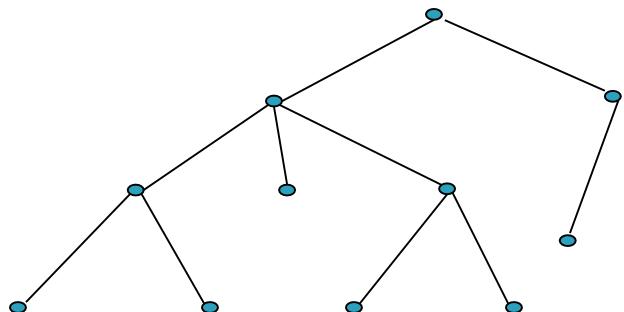
một rừng (forest): gồm nhiều cây không có đỉnh chung

**Mọi đỉnh x trong một cây mà là gốc của một cây con,
Khi xóa đỉnh x khỏi cây ta được một rừng**

Định nghĩa và các tính chất cơ bản (tt)

- Định lý 2: Nếu một cây có n đỉnh thì sẽ có $m=n-1$ cạnh

Ví dụ:



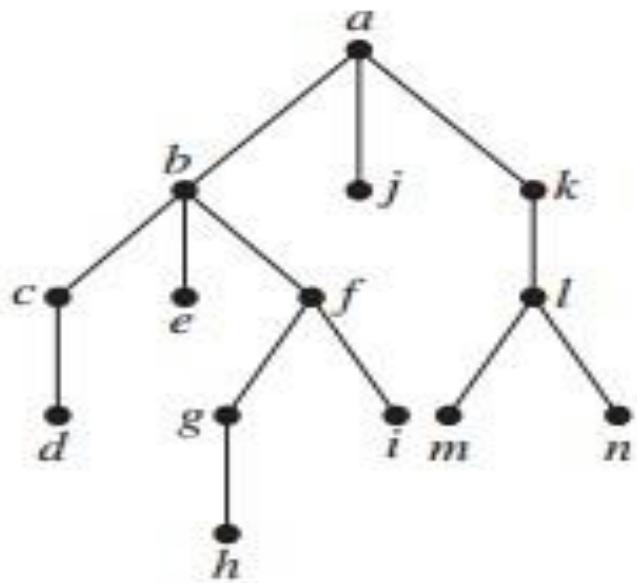
Cây có 11 đỉnh \Rightarrow có ???
Bao nhiêu cạnh

Định nghĩa và các tính chất cơ bản (tt)

Định nghĩa (độ lệch tâm): Xét cây T (tự do)

- Độ lệch tâm (eccentricity) của đỉnh v: là Khoảng cách lớn nhất từ v đến đỉnh bất kỳ trong T. Kí hiệu $E(v)$

$$E(v) = \max_{u \in T} \delta(u, v)$$



$$E(c) = ?$$

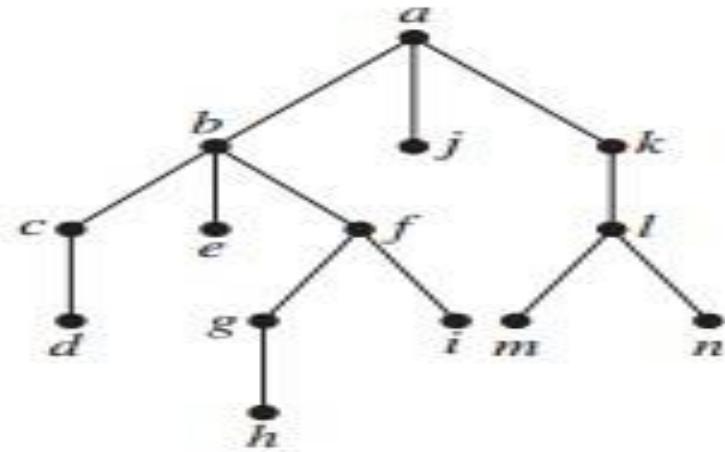
$$E(f) = ?$$

Định nghĩa và các tính chất cơ bản (tt)

Xét cây có gốc T

- Đỉnh có độ lệch tâm nhỏ nhất gọi là **tâm** (center) của T
- Độ lệch tâm của tâm gọi là **bán kính** (radius) của T

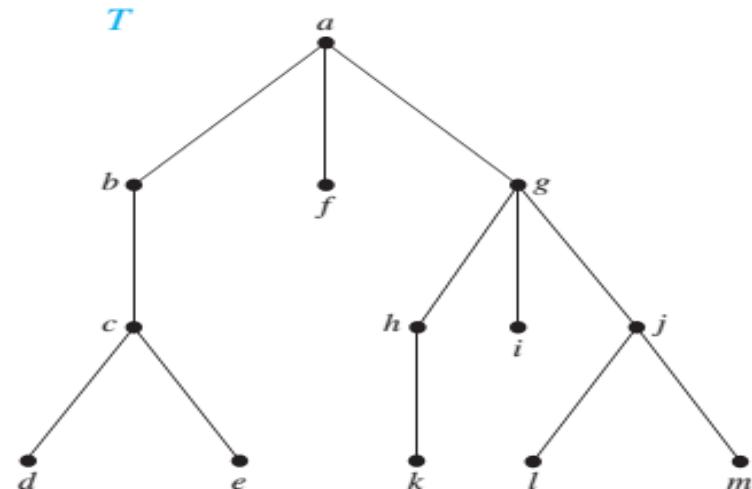
Ví dụ: Cho cây T



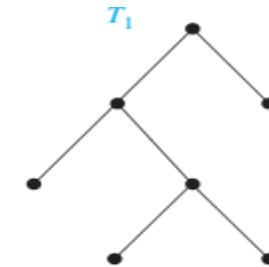
Xác định tâm của T?
Xác định bán kính của T?

Định nghĩa và các tính chất cơ bản (tt)

- Cho cây có gốc T:
 - Nếu số **con tối đa** của một đỉnh trong T là **m** và có ít nhất một đỉnh đúng **m** con thì T gọi là **cây m-phân** (**m-arytree**)
 - Nếu mọi **đỉnh trong** của T đều có đúng **m** cây **con** thì T gọi là **cây m-phân đủ** (**complete m-arytree**)
 - Cây m-phân với **m=2** gọi là **cây nhị phân**



Cây tam phân



Cây nhị phân đủ

Định nghĩa và các tính chất cơ bản (tt)

❖ Định lý 3 (Định lý Daisy Chain): T là một đồ thị có n đỉnh, các mệnh đề sau là tương đương

- (i) T là cây
- (ii) T không có chu trình và có $n-1$ cạnh
- (iii) T Liên thông và nếu hủy bất kỳ một cạnh nào trong T thì T sẽ mất tính liên thông
- (iv) Giữa 2 đỉnh bất kỳ trong T luôn tồn tại duy nhất một đường đi nối chúng
- (v) T không có chu trình, nếu thêm 1 cạnh bất kỳ nối 2 đỉnh trong T thì T sẽ có chu trình
- (vi) T liên thông và có $n-1$ cạnh

Định nghĩa và các tính chất cơ bản (tt)

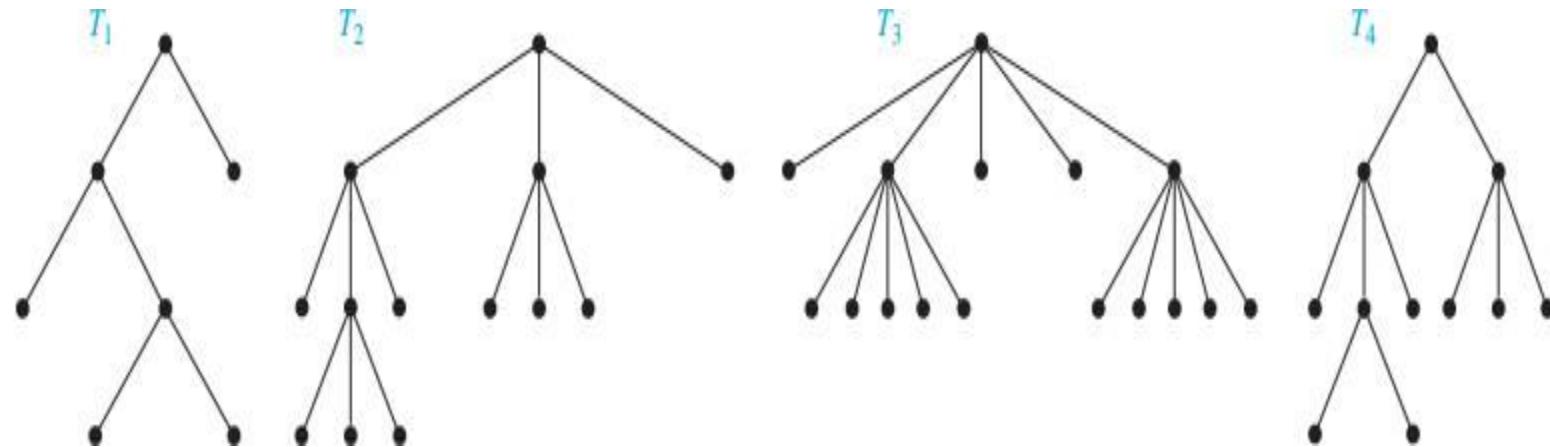
- ❖ Định lý 4: Một cây tự do có nhiều nhất 2 tâm
- ❖ Định lý 5: Một cây m-phân đầy đủ có i đỉnh trong thì có $m \cdot i + 1$ đỉnh
- Hệ luận: T là một cây m-phân đầy đủ
 - i. T có i đỉnh trong $\Rightarrow T$ có $I = (m-1)i + 1$ lá
 - ii. T có I lá $\Rightarrow T$ có $i = (I-1)/(m-1)$ đỉnh trong và $n = (ml-1)/(m-1)$ đỉnh
 - iii. T có n đỉnh $\Rightarrow T$ có $i = (n-1)/m$ đỉnh trong và $I = [(m-1)n+1]/m$ nút lá

Định nghĩa và các tính chất cơ bản (tt)

- Hệ luân: T là một cây m-phân đầy đủ
- iv. Một cây m-phân có chiều cao h thì có nhiều nhất là m^h lá
- v. Một cây m-phân có l lá thì có chiều cao $h \geq [\log_m l]$
- vi. Một cây m-phân đầy đủ, cân bằng có l lá thì có chiều cao $h = [\log_m l]$

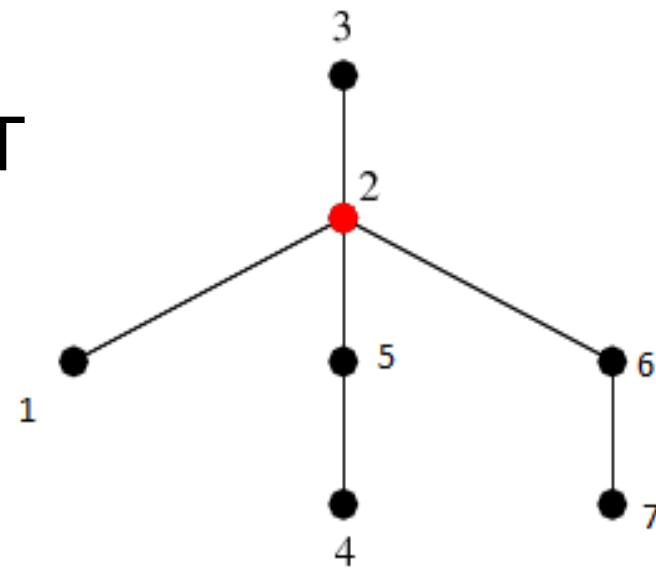
Bài tập

- Hãy xác định hệ số m của cây m-phân bên dưới



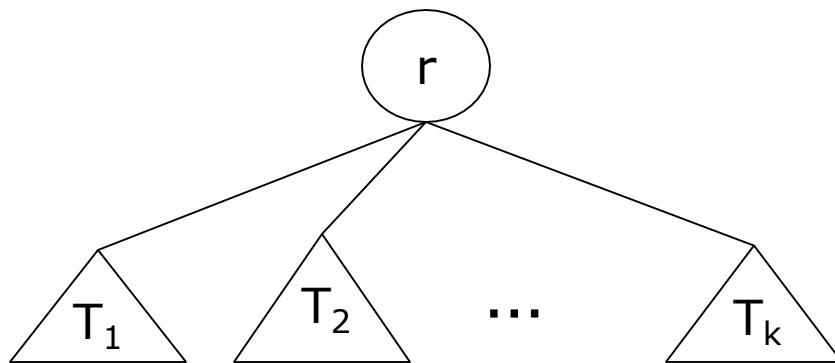
Bài tập

- ❖ Cho 1 cây T như hình vẽ, Xác định các yếu tố của cây T:
 - Cây có bao nhiêu cạnh
 - Xác định độ lệch tâm của tất cả các Đỉnh
 - Xác định đỉnh Tâm của T
 - Bán kính của T
 - Đường kính T



2. Các phương pháp duyệt cây

Xét cây có gốc T , gọi $T_{r1}, T_{r2}, \dots, T_{rk}$ lần lượt là các cây con của nút r theo thứ tự từ trái qua phải

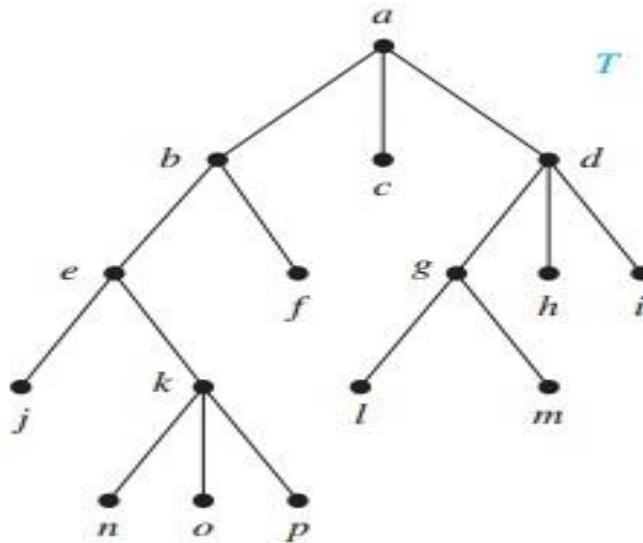


2. Các phương pháp duyệt cây

2.1. Duyệt cây theo thứ tự trước (preorder : Root –Left Subtree – right Subtree)

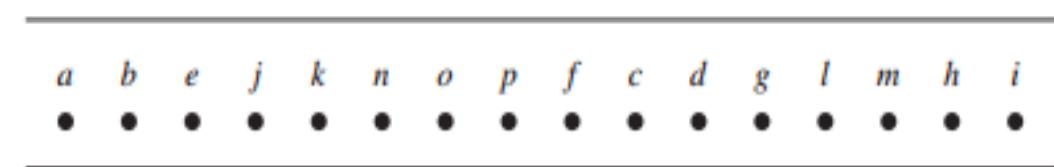
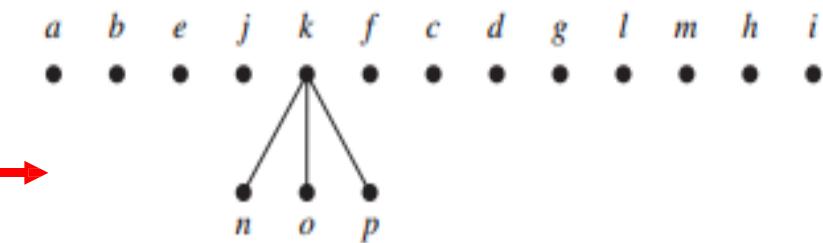
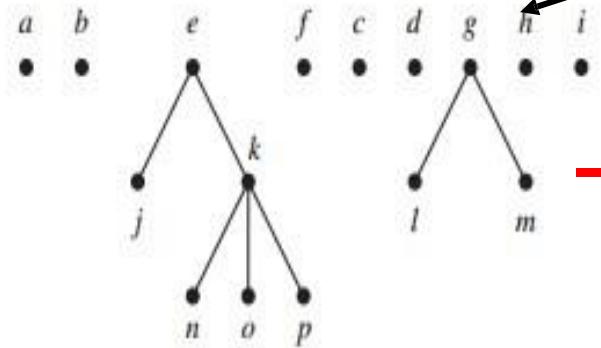
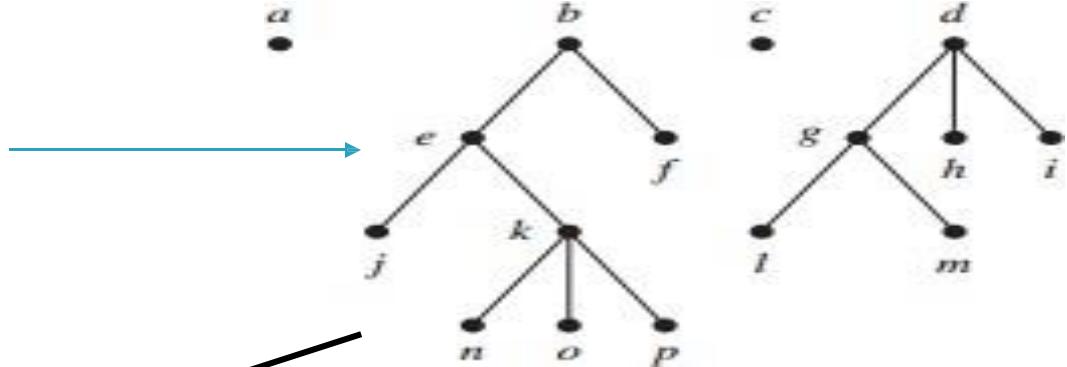
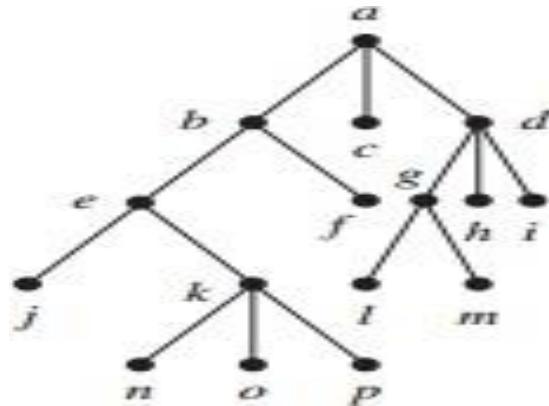
- Thăm gốc r của T
- Đệ quy: Duyệt từng cây con lần lượt từ T_1 đến T_{rk} theo thứ tự trước

Ví dụ:



Kết quả duyệt theo Preorder?

2.1. Duyệt cây theo thứ tự trước (preorder : Root – Left Subtree – right Subtree)



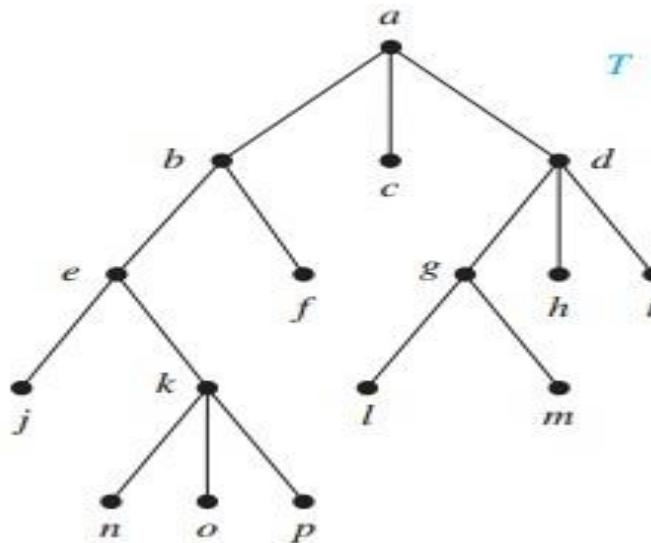
ABEJKNO_PFCDGLMHI

2. Các phương pháp duyệt cây

2.2. Duyệt cây theo thứ tự giữa (inorder) (**left subtree – root – right subtree**)

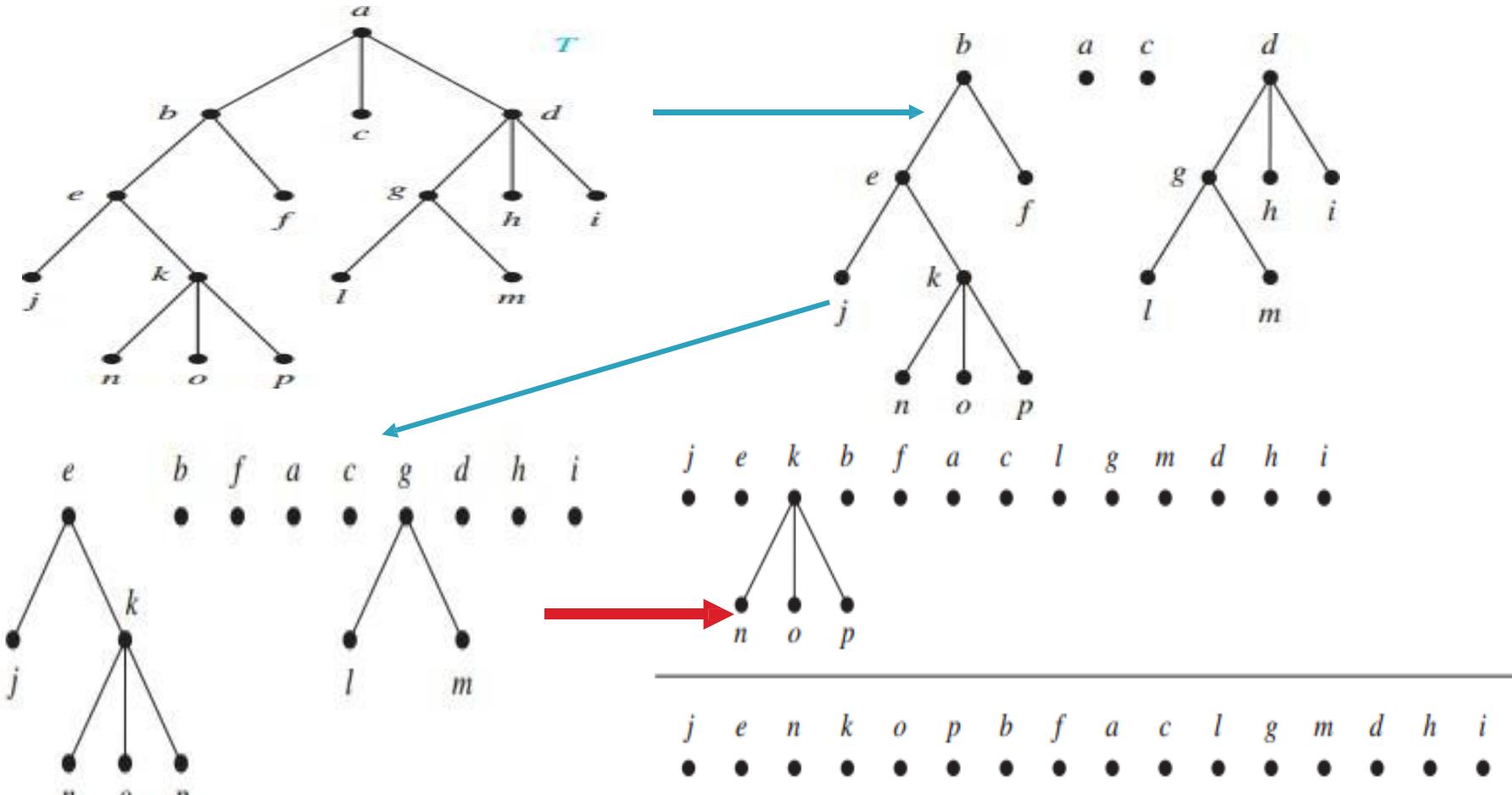
- Duyệt T_{r1} theo thứ tự giữa
- Thăm r
- Duyệt T_{r2} theo thứ tự giữa,..., duyệt T_{rk} theo thứ tự giữa

Ví dụ:



Kết quả duyệt theo Inorder?

2.2. Duyệt cây theo thứ tự giữa (inoder) (left subtree – root – right subtree)

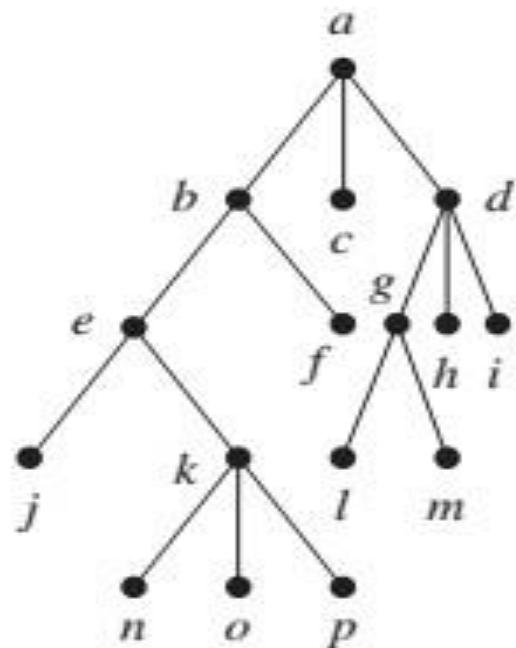


2. Các phương pháp duyệt cây

2.3. Duyệt cây theo thứ tự sau (postorder : left subtree – right subtree - root)

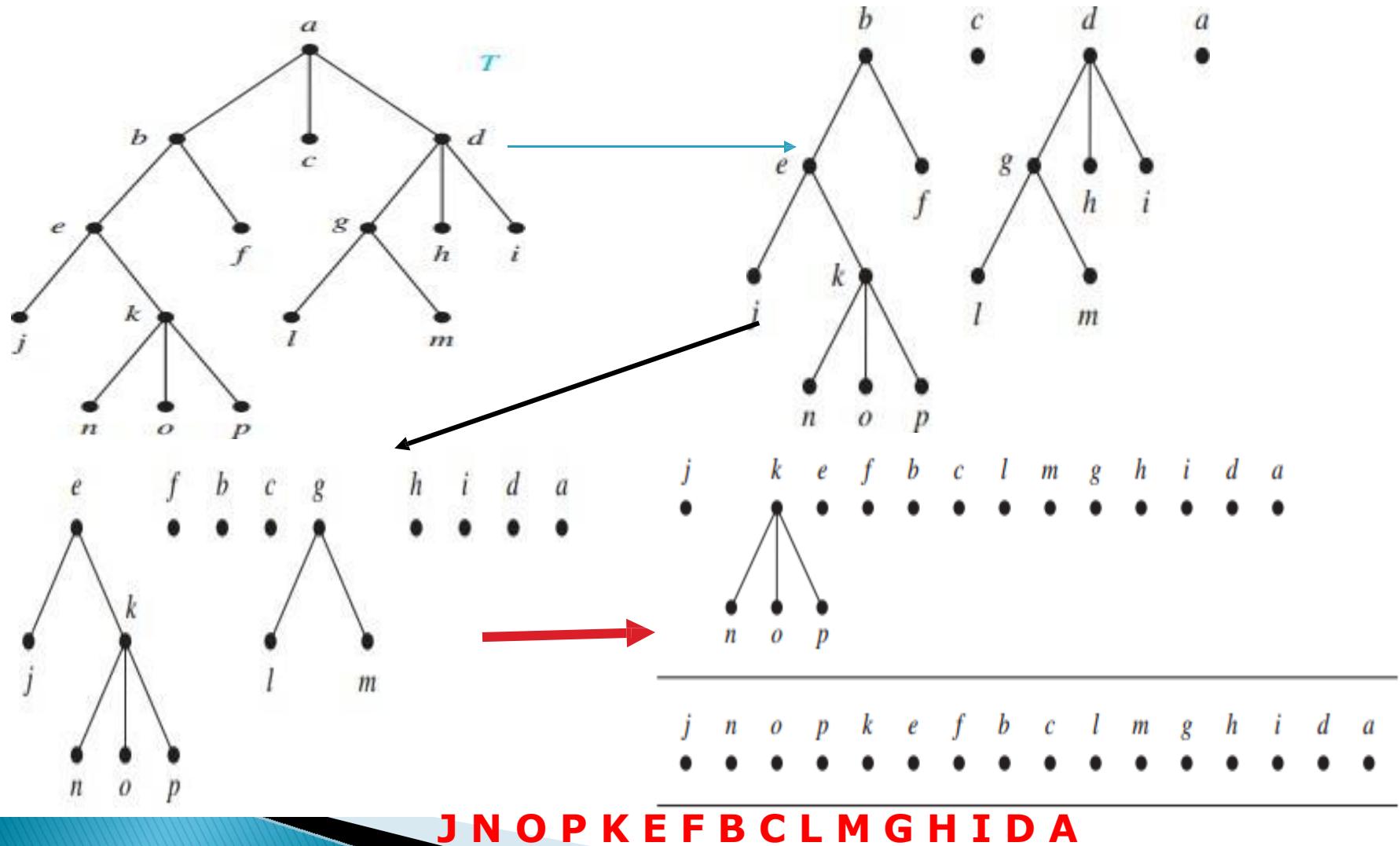
- Duyệt T_{r_1} theo postorder, duyệt T_{r_2} theo postorder, ..., duyệt T_{r_k} theo postorder
- Thăm r

Ví dụ:



Kết quả duyệt theo postorder?

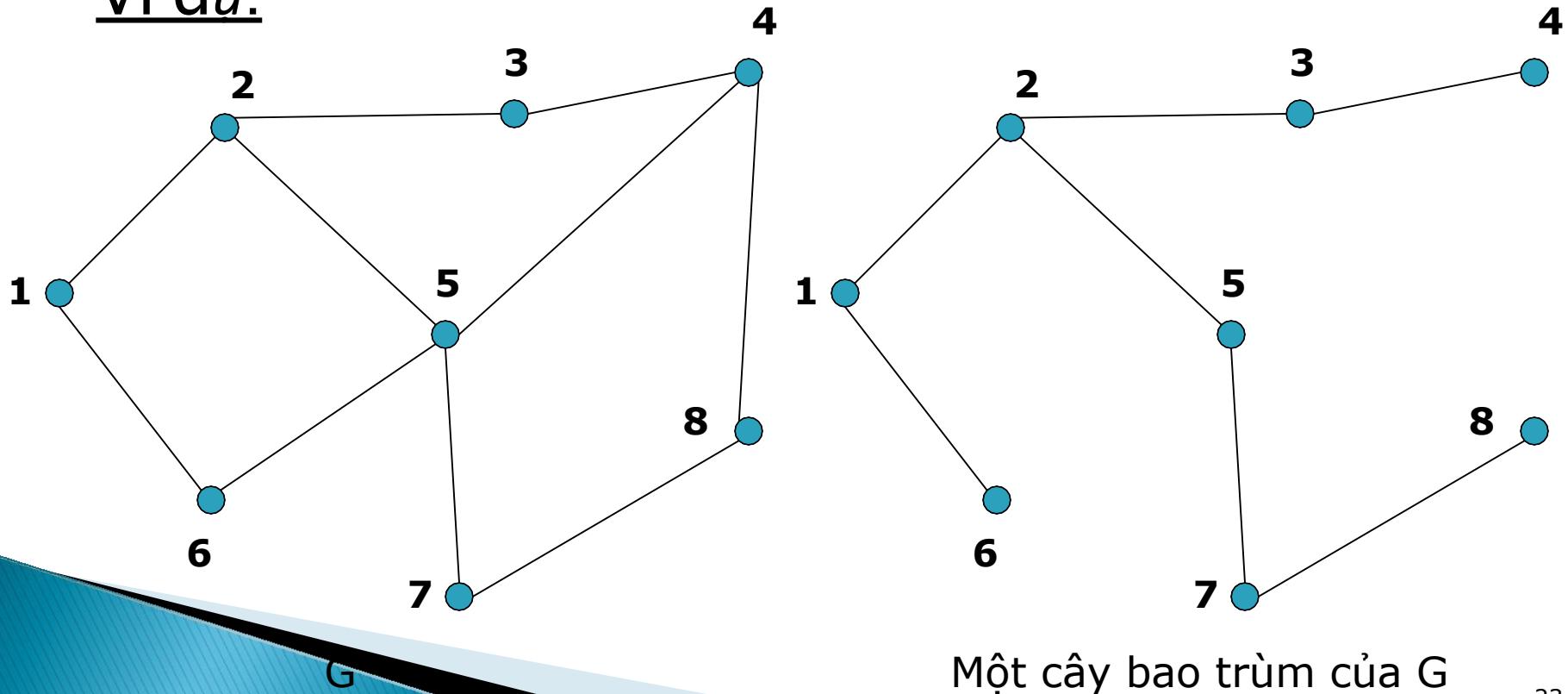
2.3. Duyệt cây theo thứ tự sau (postorder : left subtree -right subtree – root)



Cây bao trùm(spanning tree)

- ❖ Cho đồ thị vô hướng G . Cây T gọi là một cây bao trùm của G nếu $T \leq G$ (T là cây khung của G) và T chứa mọi đỉnh của G

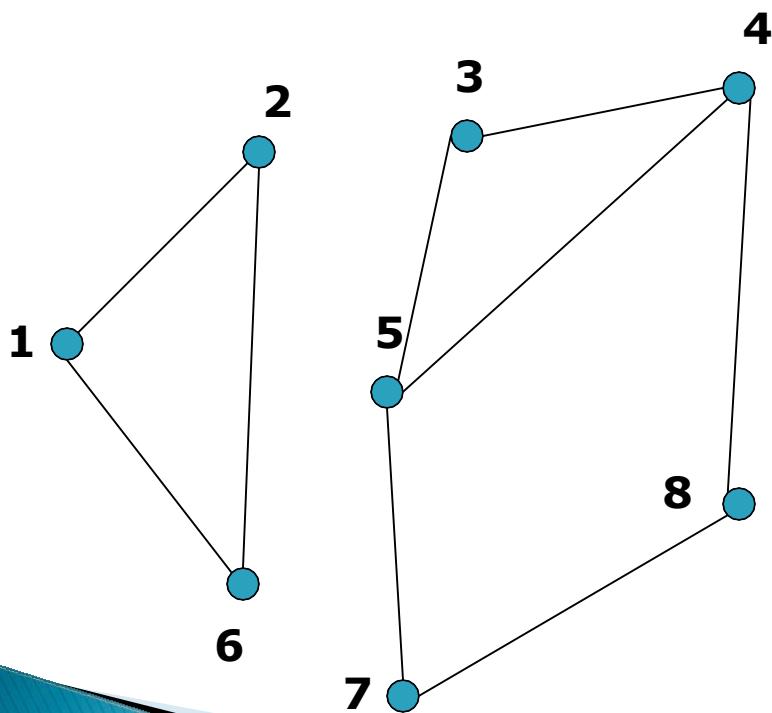
Ví dụ:



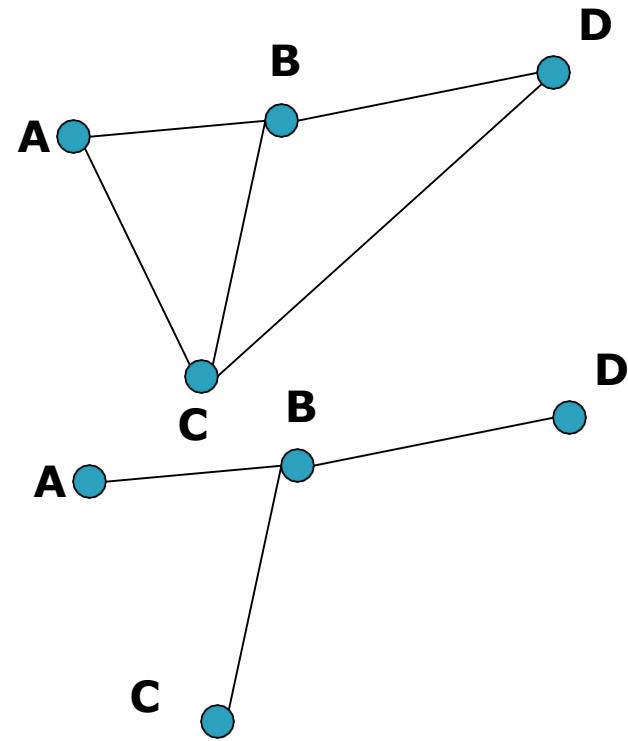
Một cây bao trùm của G

Cây bao trùm(spanning tree)

- ❖ Định lý: Đồ thị G có cây bao trùm \Leftrightarrow G liên thông



G không liên thông \rightarrow G không có cây bao trùm



H liên thông \Rightarrow H có cây bao trùm

Tìm cây bao trùm

Hai phương pháp

- ❖ Tìm theo chiều sâu (DFS: Depth First Search)
- ❖ Tìm theo chiều rộng (BFS: Breadth First Search)

Tìm cây bao trùm theo phương pháp – DFS (Depth First Search)

Bước 1: Chọn một đỉnh bất kỳ v của G làm điểm xuất phát (gốc) của T .

Bước 2: Xác định một đường đi sơ cấp xuất phát từ v qua các đỉnh $\in G$ nhưng $\notin T$, cho đến khi không thể đi tiếp. Gọi k là đỉnh kết thúc đường đi. Đặt đường đi này vào T rồi quay trở về đặt đỉnh liền trước k làm điểm xuất phát. Lặp lại thủ tục này cho đến khi mọi đỉnh của G đều nằm trong T .

Tìm cây bao trùm theo phương pháp – DFS (Depth First Search)

DFS(G: Liên thông với các đỉnh v_1, v_2, \dots, v_n) {

T=Cây chỉ gồm 1 đỉnh v_1 visit(v_1)

}

Visit(v: đỉnh của G) {

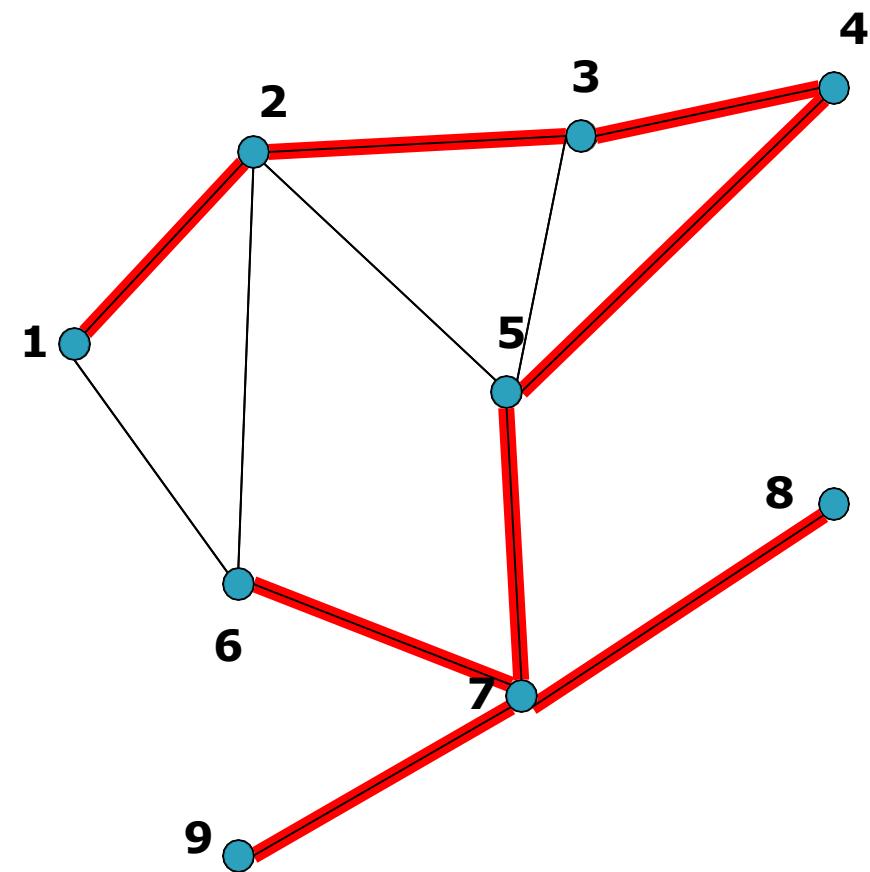
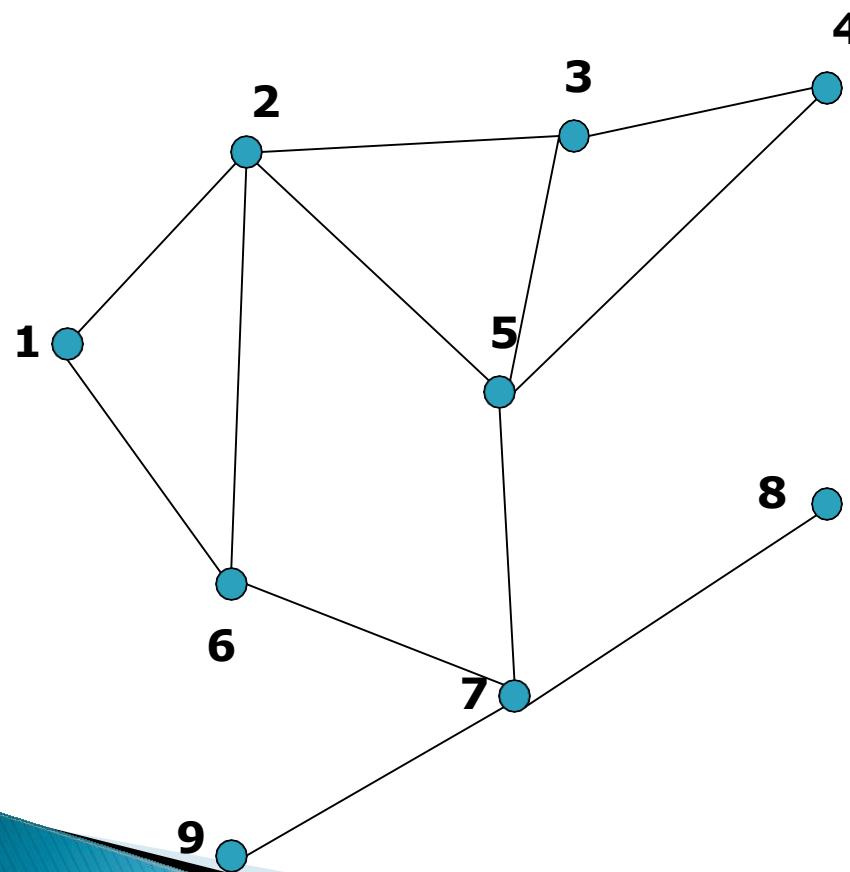
– For (mỗi đỉnh w kề với v nhưng chưa có trong T)

– Thêm đỉnh w và cạnh (v, w) vào T

visit(w);

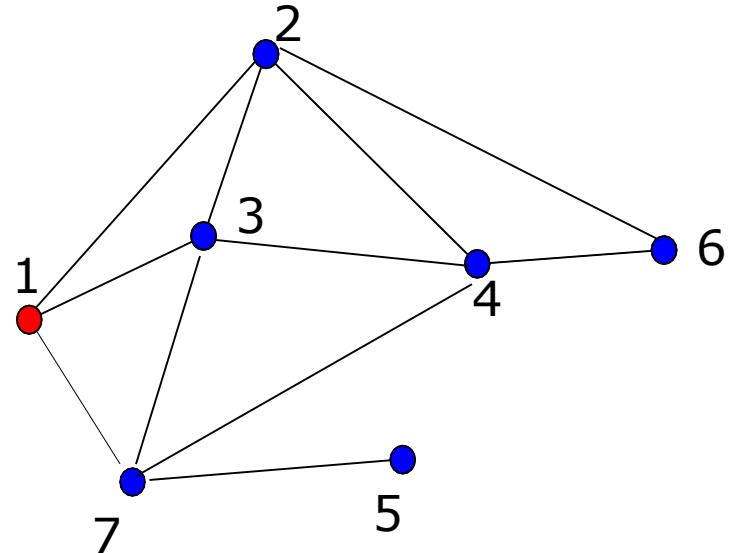
}

Tìm cây bao trùm theo phương pháp – DFS (Depth First Search)



Tìm cây bao trùm theo phương pháp - DFS (Depth First Search)

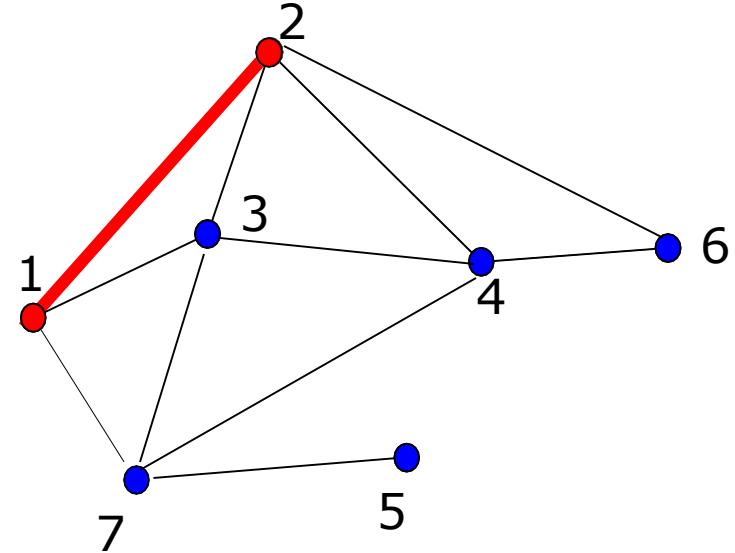
	1	2	3	4	5	6	7
1	0	1	1	0	0	0	1
2	1	0	1	1	0	1	0
3	1	1	0	1	0	0	1
4	0	1	1	0	0	1	1
5	0	0	0	0	0	0	1
6	0	1	0	1	0	0	0
7	1	0	1	1	1	0	0



$v=1; T = \langle V_T, E_T \rangle$ với $V_T = \{1\}$, $E_T = \emptyset$

Tìm cây bao trùm theo phương pháp - DFS (Depth First Search)

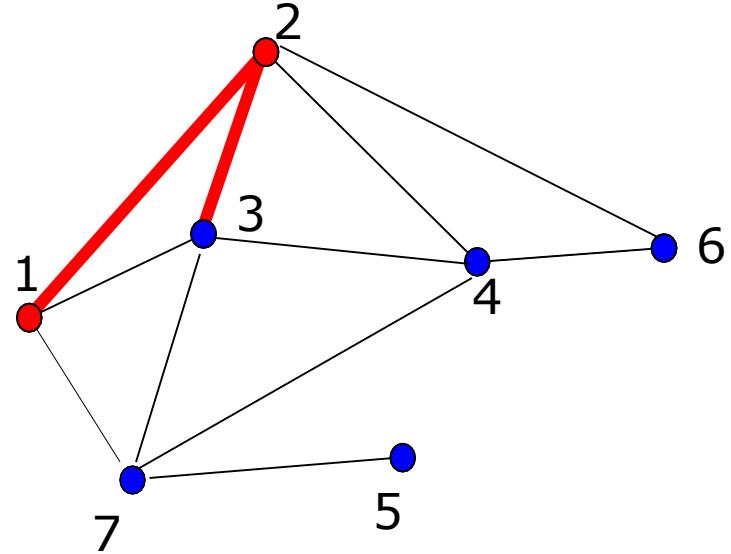
	1	2	3	4	5	6	7
1	0	1	1	0	0	0	1
2	1	0	1	1	0	1	0
3	1	1	0	1	0	0	1
4	0	1	1	0	0	1	1
5	0	0	0	0	0	0	1
6	0	1	0	1	0	0	0
7	1	0	1	1	1	0	0



$$V=1, w=2, a_{12} \neq 0 \wedge 2 \notin V_T, V_T = \{1, 2\}, E_T = \{(1, 2)\}$$

Tìm cây bao trùm theo phương pháp - DFS (Depth First Search)

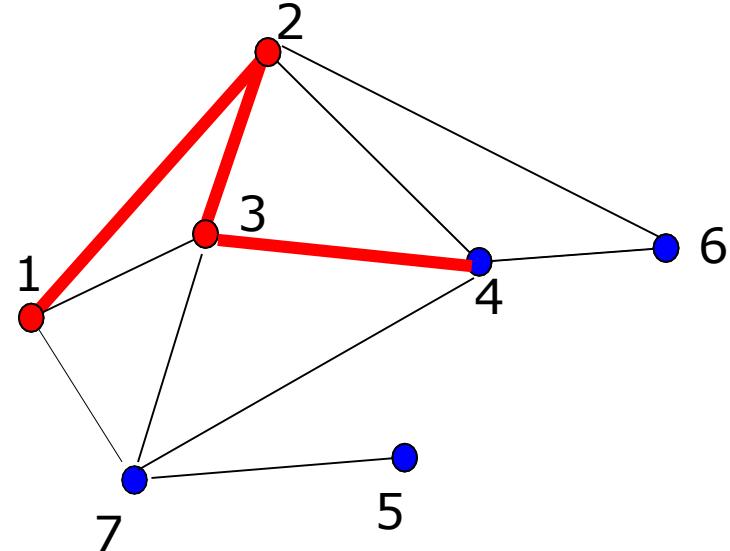
	1	2	3	4	5	6	7
1	0	1	1	0	0	0	1
2	1	0	1	1	0	1	0
3	1	1	0	1	0	0	1
4	0	1	1	0	0	1	1
5	0	0	0	0	0	0	1
6	0	1	0	1	0	0	0
7	1	0	1	1	1	0	0



$V=2, w=3, a_{23} \neq 0 \wedge 3 \notin V_T, V_T = \{1, 2, 3\},$
 $E_T = \{(1, 2), (2, 3)\}$

Tìm cây bao trùm theo phương pháp - DFS (Depth First Search)

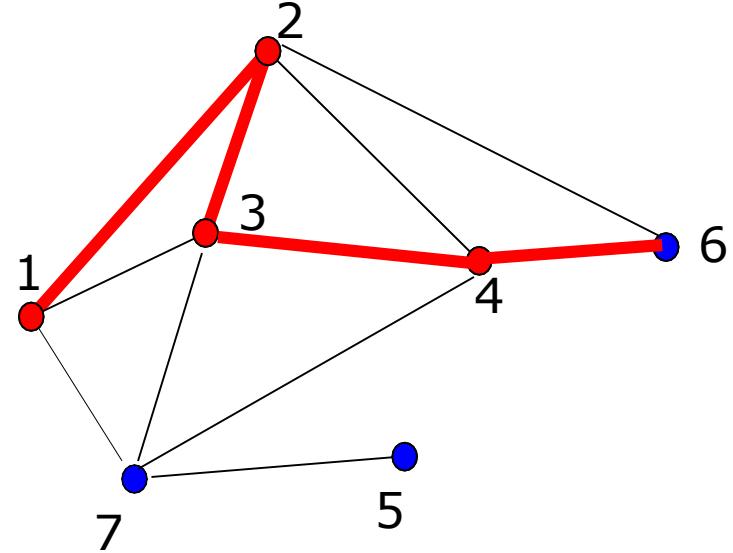
	1	2	3	4	5	6	7
1	0	1	1	0	0	0	1
2	1	0	1	1	0	1	0
3	1	1	0	1	0	0	1
4	0	1	1	0	0	1	1
5	0	0	0	0	0	0	1
6	0	1	0	1	0	0	0
7	1	0	1	1	1	0	0



$V=3, w=4, a_{34} \neq 0 \wedge 4 \notin V_T, V_T = \{1, 2, 3, 4\},$
 $E_T = \{(1, 2), (2, 3), (3, 4)\}$

Tìm cây bao trùm theo phương pháp - DFS (Depth First Search)

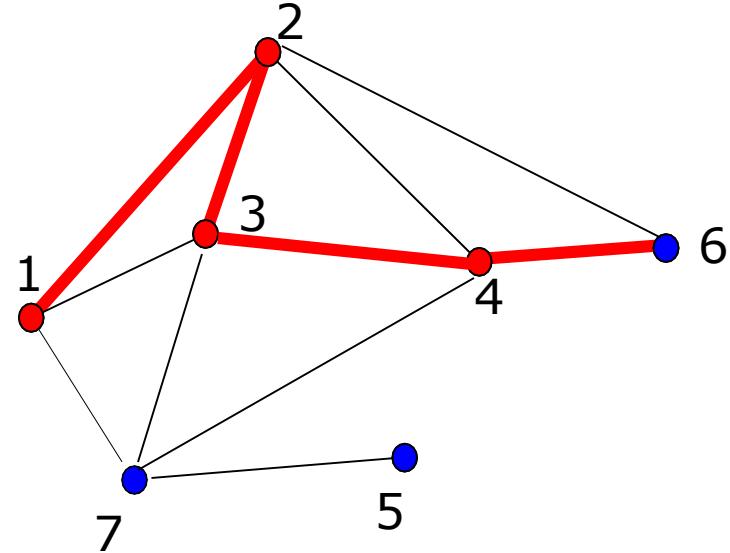
	1	2	3	4	5	6	7
1	0	1	1	0	0	0	1
2	1	0	1	1	0	1	0
3	1	1	0	1	0	0	1
4	0	1	1	0	0	1	1
5	0	0	0	0	0	0	1
6	0	1	0	1	0	0	0
7	1	0	1	1	1	0	0



$V=4$, $w=6$, $a_{46} \neq 0 \wedge 6 \notin V_T$, $V_T = \{1, 2, 3, 4, 6\}$,
 $E_T = \{(1, 2), (2, 3), (3, 4), (4, 6)\}$

Tìm cây bao trùm theo phương pháp - DFS (Depth First Search)

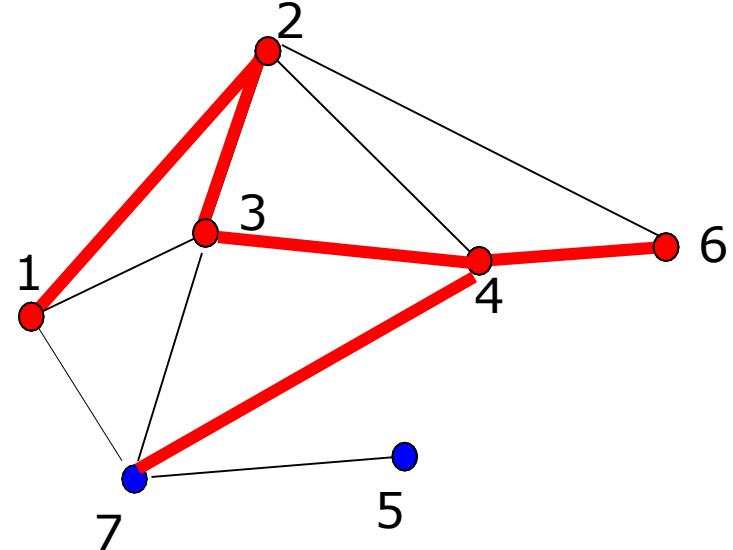
	1	2	3	4	5	6	7
1	0	1	1	0	0	0	1
2	1	0	1	1	0	1	0
3	1	1	0	1	0	0	1
4	0	1	1	0	0	1	1
5	0	0	0	0	0	0	1
6	0	1	0	1	0	0	0
7	1	0	1	1	1	0	0



$V=4, W=7, a_{47} \neq 0 \wedge 7 \notin V_T, V_T = \{1, 2, 3, 4, 6, 7\},$
 $E_T = \{(1, 2), (2, 3), (3, 4), (4, 6), (4, 7)\}$

Tìm cây bao trùm theo phương pháp - DFS (Depth First Search)

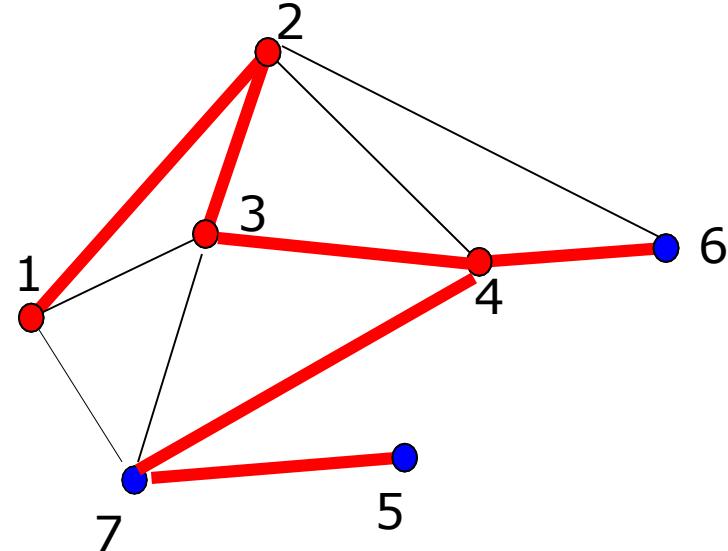
	1	2	3	4	5	6	7
1	0	1	1	0	0	0	1
2	1	0	1	1	0	1	0
3	1	1	0	1	0	0	1
4	0	1	1	0	0	1	1
5	0	0	0	0	0	0	1
6	0	1	0	1	0	0	0
7	1	0	1	1	1	0	0



$V=4, W=7, a_{47} \neq 0 \wedge 7 \notin V_T, V_T = \{1, 2, 3, 4, 6, 7\},$
 $E_T = \{(1,2), (2,3), (3,4), (4,6), (4,7)\}$

Tìm cây bao trùm theo phương pháp - DFS (Depth First Search)

	1	2	3	4	5	6	7
1	0	1	1	0	0	0	1
2	1	0	1	1	0	1	0
3	1	1	0	1	0	0	1
4	0	1	1	0	0	1	1
5	0	0	0	0	0	0	1
6	0	1	0	1	0	0	0
7	1	0	1	1	1	0	0



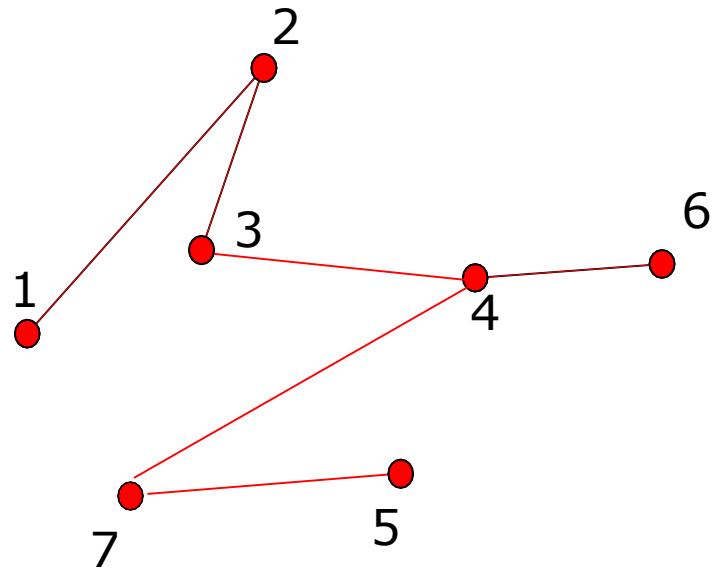
$V=7, W=5, a_{75} \neq 0 \wedge 5 \notin V_T, V_T = \{1, 2, 3, 4, 6, 7, 5\},$
 $E_T = \{(1,2), (2,3), (3,4), (4,6), (4,7), (7,5)\}$

Mọi cạnh của G đều có trong T, dừng

Tìm cây bao trùm theo phương pháp - DFS (Depth First Search)

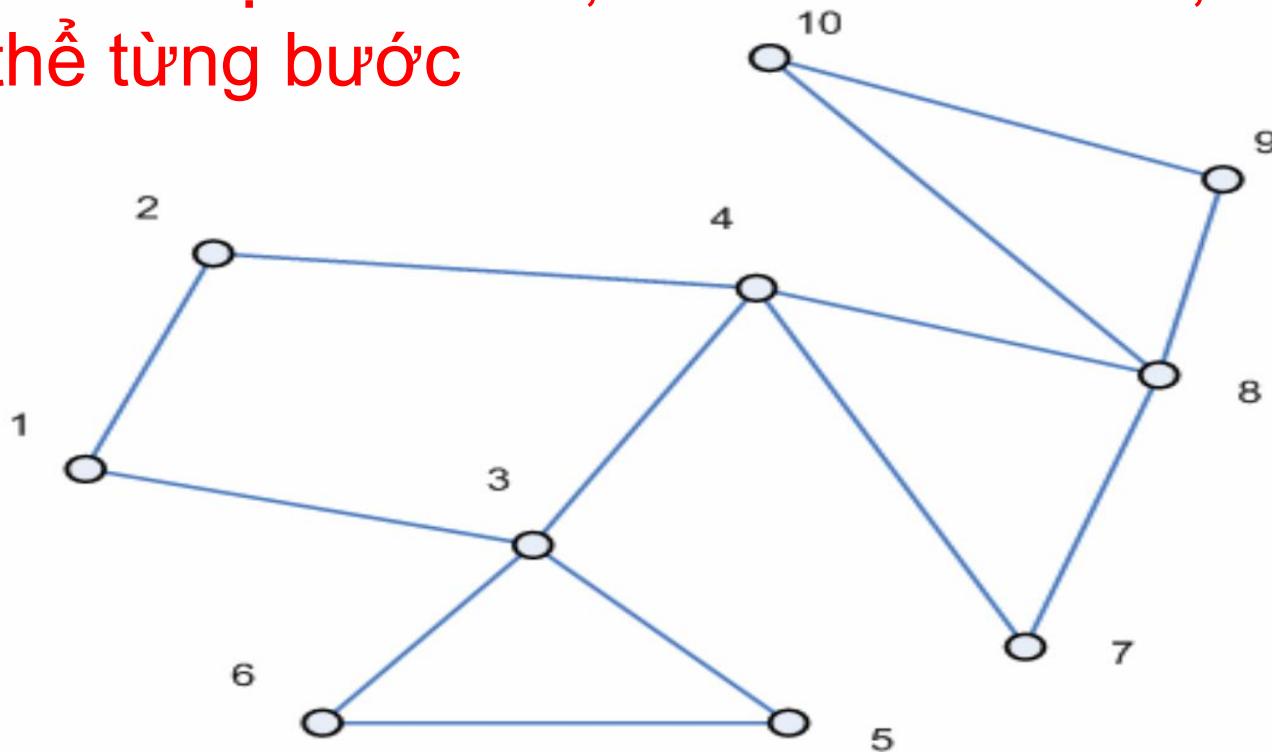
	1	2	3	4	5	6	7
1	0	1	1	0	0	0	1
2	1	0	1	1	0	1	0
3	1	1	0	1	0	0	1
4	0	1	1	0	0	1	1
5	0	0	0	0	0	0	1
6	0	1	0	1	0	0	0
7	1	0	1	1	1	0	0

Cây bao trùm tìm được



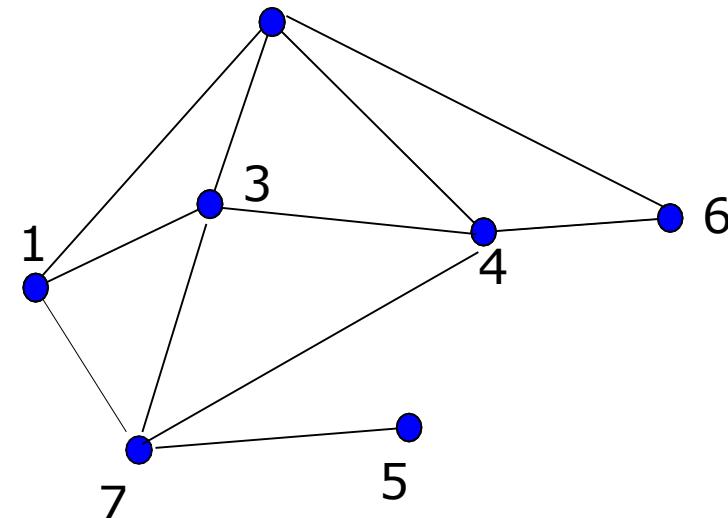
Tìm cây bao trùm theo phương pháp - DFS (Depth First Search)

Bài tập: Tìm cây bao trùm theo phương pháp DFS cho đồ thị bên dưới, bắt đầu từ đỉnh 1, ghi cụ thể từng bước



Tìm cây bao trùm theo phương pháp – BFS (Breadth First Search)

- 1) Chọn 1 đỉnh bất kỳ của G làm đỉnh xuất phát (gốc) của T.
- 2) Đặt mọi cạnh nối gốc với 1 đỉnh $\notin T$ vào T.
Lần lượt xét từng đỉnh con trực tiếp của gốc.
Xem đỉnh này là gốc mới, lặp lại thủ tục này
cho đến khi mọi đỉnh của G đều nằm trong T.



Tìm cây bao trùm theo phương pháp - BFS (Breadth First Search)

BFS(G: liên thông với tập đỉnh v_1, v_2, \dots, v_n) {

 T:= Cây chỉ gồm 1 đỉnh v_1 ;

 Q={v} // Queue: các đỉnh chưa xử lý
 while ($Q \neq \emptyset$){

 v = Q.Remove();

 for (mỗi đỉnh w kề với v)

 if $w \notin Q$ and $w \notin T\{$

 Q.Add(w)

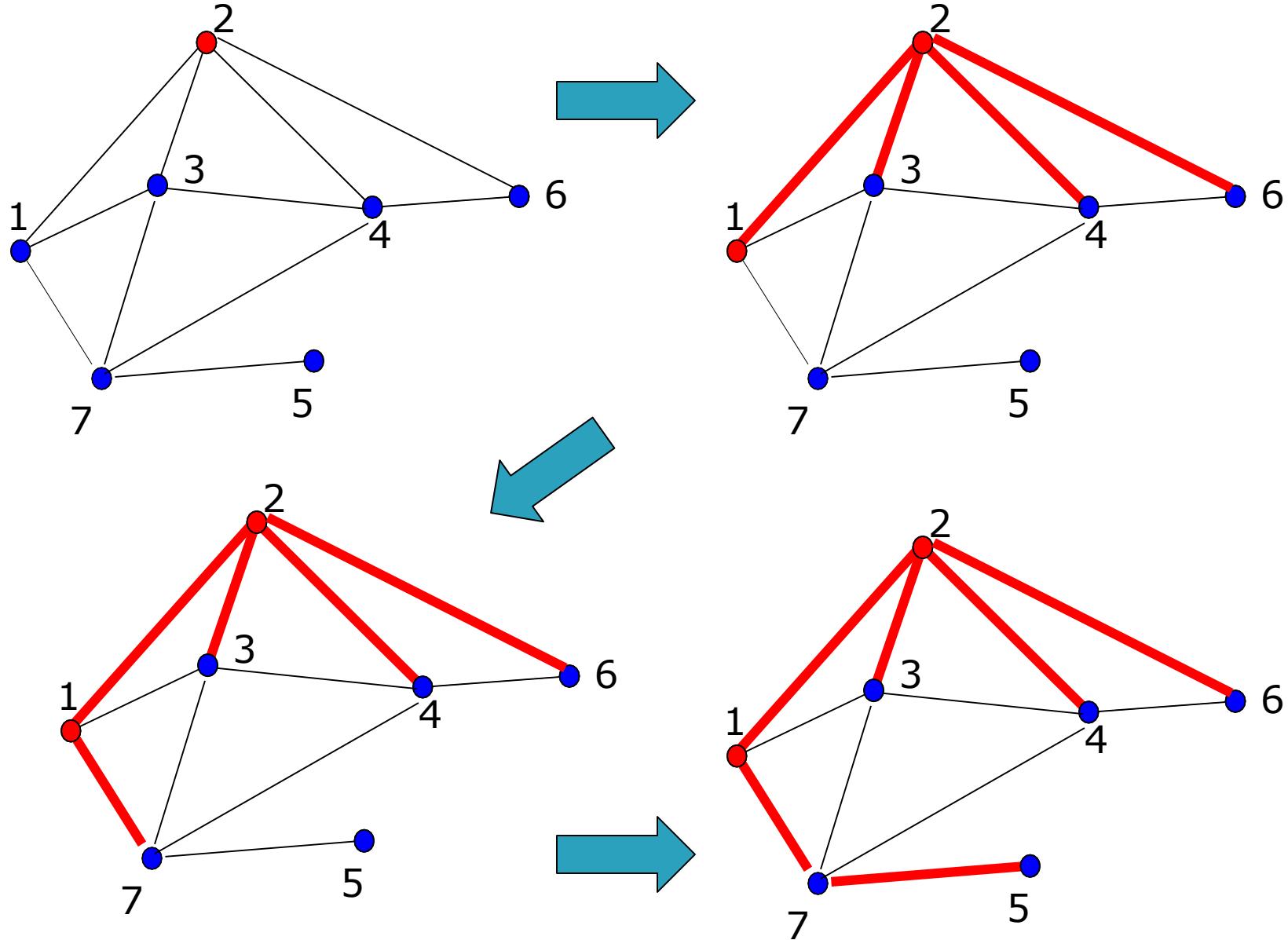
 Thêm cạnh (v, w) vào T

 }

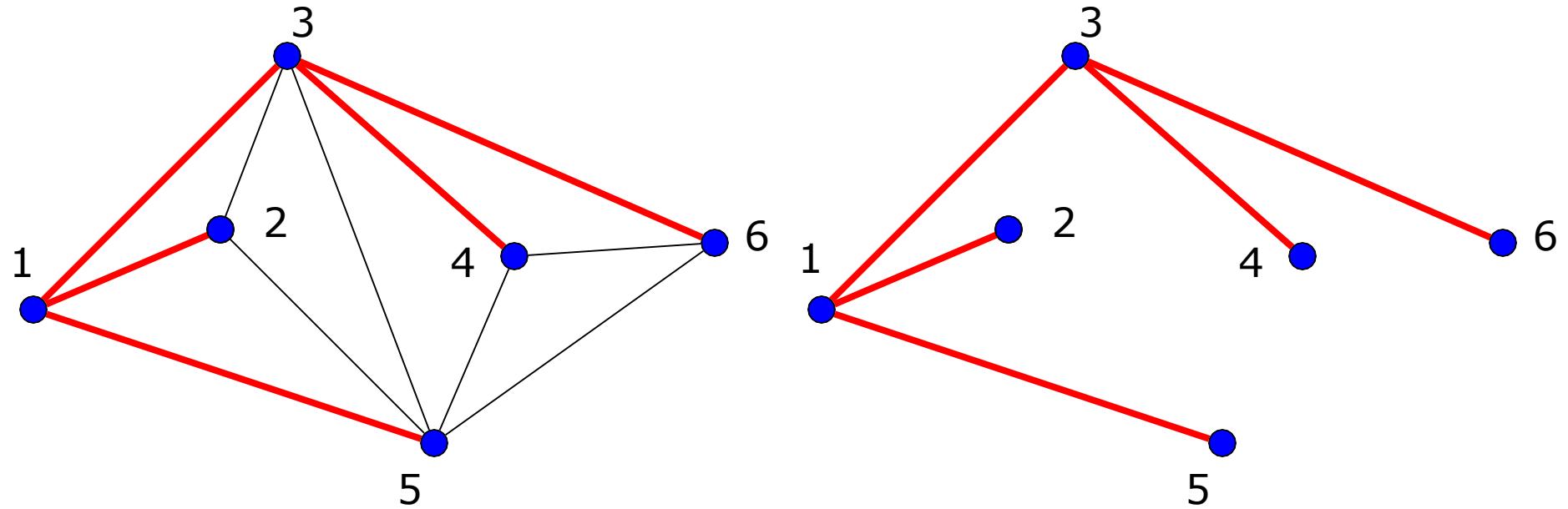
}

}

Tìm cây bao trùm theo phương pháp - BFS (Breadth First Search)

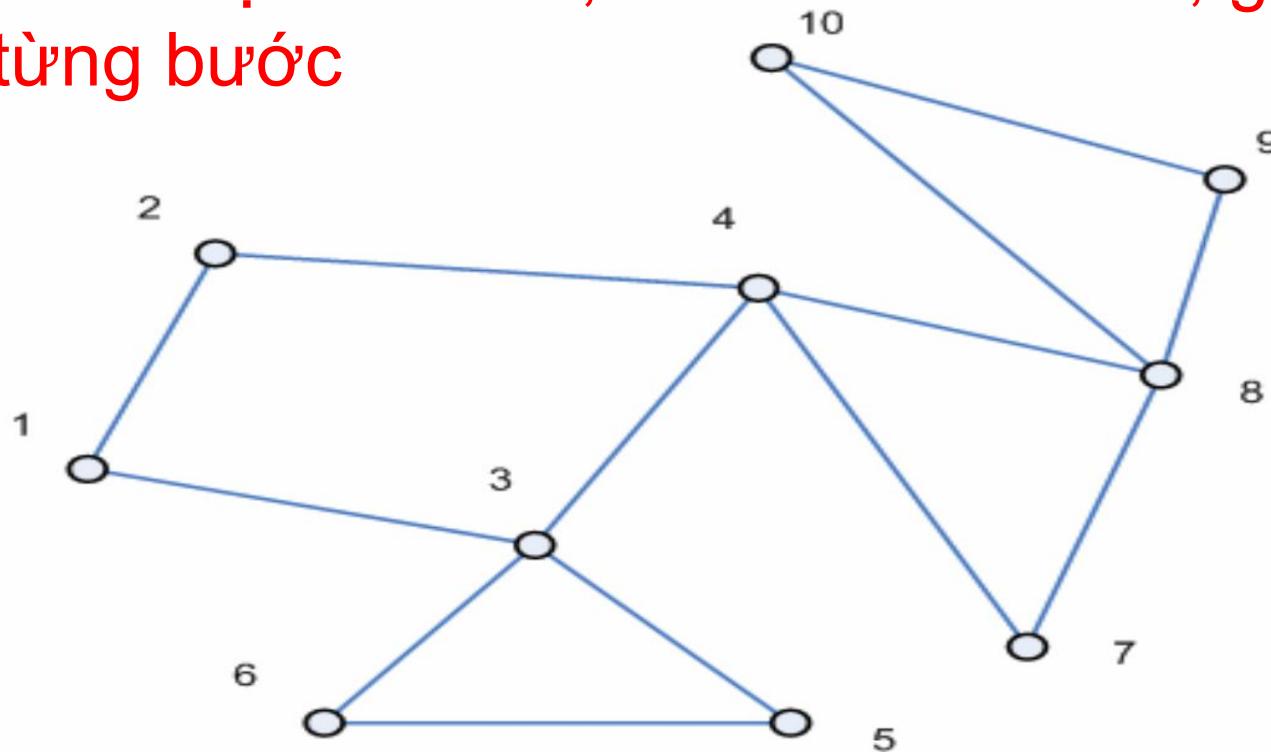


Tìm cây bao trùm theo phương pháp - BFS (Breadth First Search)



Tìm cây bao trùm theo phương pháp – BFS (Breadth First Search)

Bài tập: Tìm cây bao trùm theo phương pháp BFS cho đồ thị bên dưới, bắt đầu từ đỉnh 1, ghi cụ thể từng bước



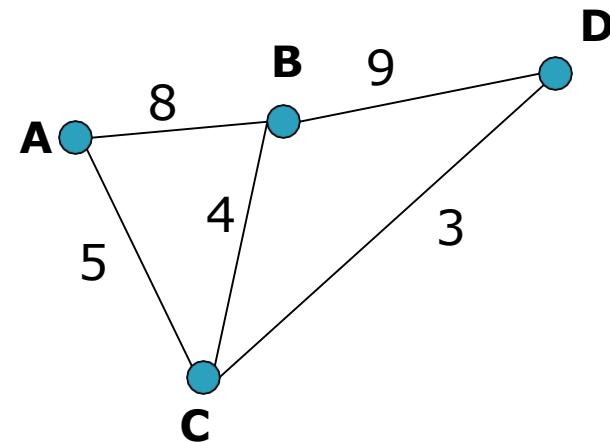
Cây bao trùm nhỏ nhất

- ❖ Đồ thị có trọng số: Là đồ thị trong đó mỗi cạnh (cung) được gán thêm một số thực gọi là trọng số (weight) của cạnh (cung)

Kí hiệu:

$c(e)$: Trọng số của cạnh e

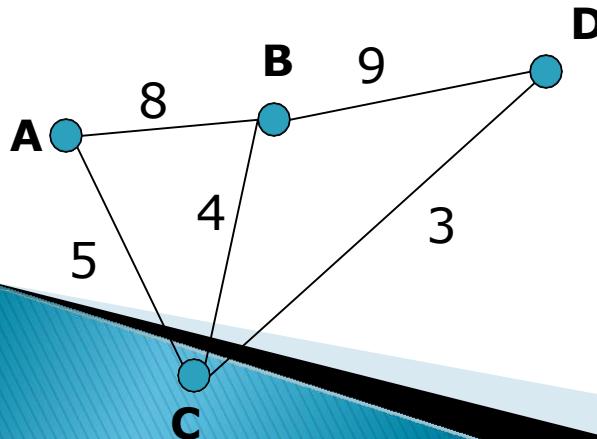
$c(G)$: Trọng số của đồ thị G



Cây bao trùm nhỏ nhất

- ❖ Ma trận kề của đồ thị có trọng số: Cho $G = \langle V, E \rangle$ có trọng số, ma trận kề trọng số của G là ma trận A có kích cỡ $|V| \times |V|$ trong đó mỗi phần tử a_{ij} có giá trị như sau:

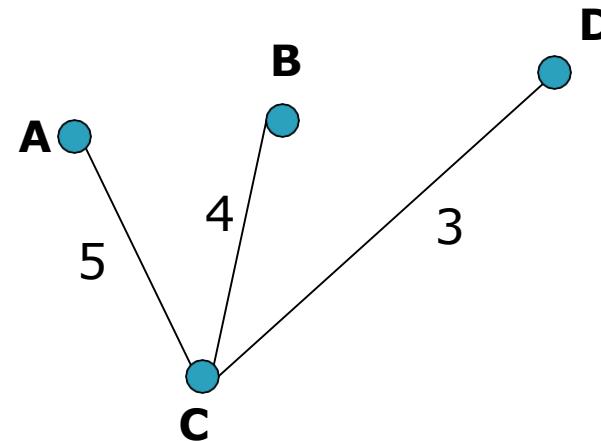
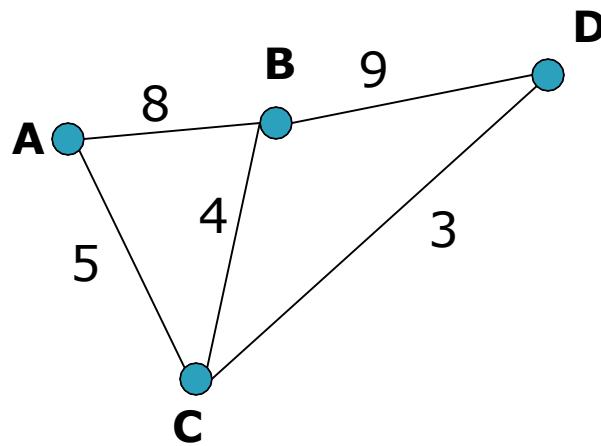
$$a_{ij} = \begin{cases} \text{Trọng số của cạnh/cung } (v_i, v_j) \text{ nếu } (v_i, v_j) \in E \\ \infty \quad \text{Nếu } (v_i, v_j) \notin E \end{cases}$$



	A	B	C	D
A	∞	8	5	∞
B	8	∞	4	9
C	5	4	∞	3
D	∞	9	3	∞

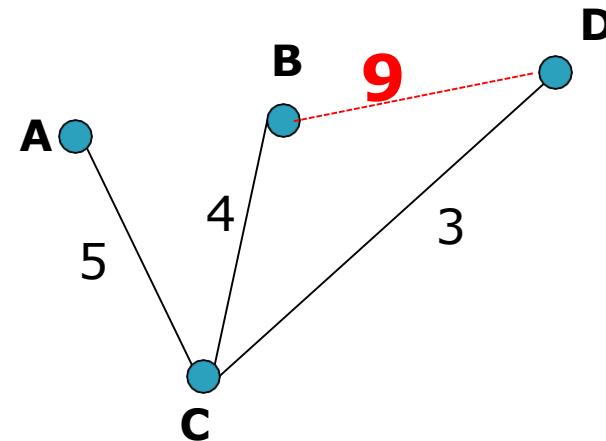
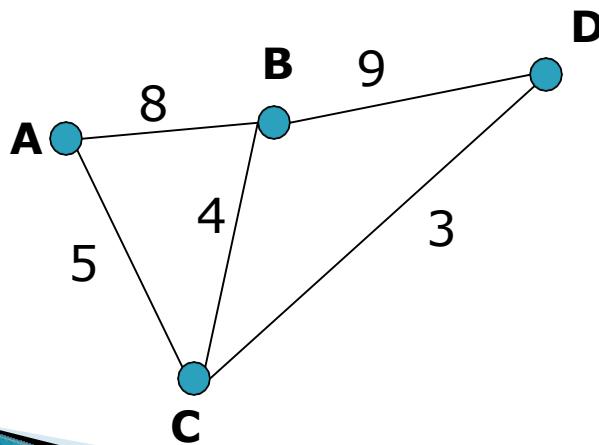
Cây bao trùm nhỏ nhất

- ❖ Bài toán: Cho G là đồ thị liên thông, có trọng số. Hãy tìm một cây bao trùm của G có trọng số nhỏ nhất



Cây bao trùm nhỏ nhất

- ❖ Định lý: Cho T là một cây bao trùm của đồ thị có trọng số G liên thông. T là một cây bao trùm tối thiểu nếu và chỉ nếu mỗi cạnh $e \notin T$ đều có trọng số lớn nhất trên chu trình tạo bởi e và các cạnh của T



Thuật toán tìm cây bao trùm nhỏ nhất: Thuật toán KRUSKAL

❖ Các bước của thuật toán KRUSKAL

- **Bước 1:** Chọn một cạnh bất kỳ có trọng số nhỏ nhất, đặt nó vào cây khung.
- **Bước 2:** Lần lượt ghép vào cây các cạnh có trọng số nhỏ nhất mà không tạo ra chu trình trong cây.
- **Bước 3:** Thuật toán dừng lại khi $(n - 1)$ cạnh được ghép vào cây.

Thuật toán tìm cây bao trùm nhỏ nhất: Thuật toán KRUSKAL

Kruskal(G: đồ thi liên thông có trọng số)

begin

 T := \emptyset ;

 E = E_G; // E_G là tập cạnh của G

 While |E_T| < n - 1 and (E ≠ \emptyset)

 begin Chọn e là cạnh độ dài nhỏ nhất trong E, E := E \ {e}

 If (T ∪ {e} không chứa chu trình) then

 // Kết nạp cạnh e vào cây khung nhỏ nhất

 T := T ∪ {e}

 end

 If (|E_T| < (n - 1)) then

 Đồ thi không liên thông.

 else

return T;

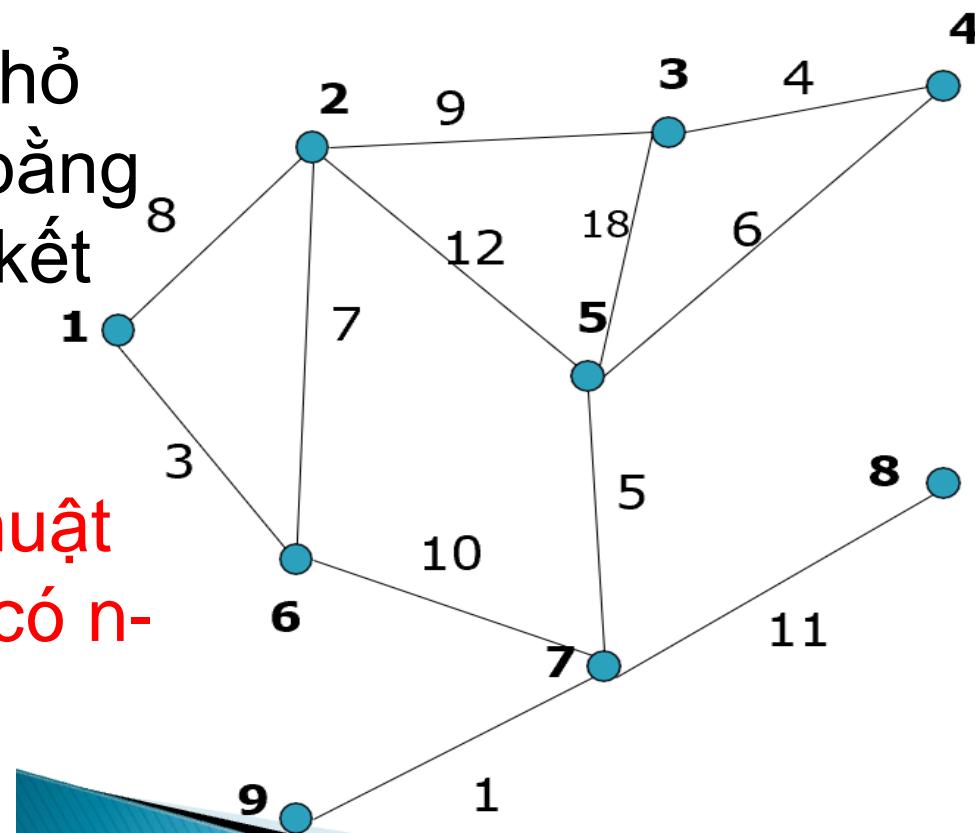
end

Thuật toán tìm cây bao trùm nhỏ nhất: Thuật toán KRUSKAL

Ví dụ: Tìm cây bao trùm nhỏ nhất của đồ thị bên dưới bằng thuật toán Kruskal, ghi ra kết quả từng bước chạy.

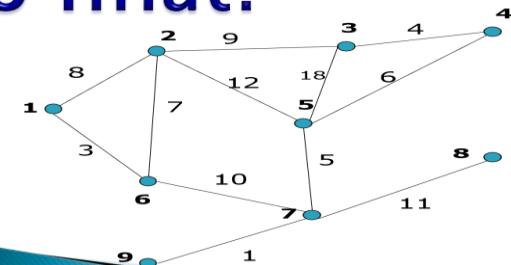
- G gồm có 9 đỉnh
- Đồ thị G có n phần tử. Thuật toán Kruskal sẽ dùng khi có $n - 1$ trong tập hợp T

- $n = 9$
- Vậy số cạnh trong tập hợp T: $n - 1 = 9 - 1 = 8(*)$



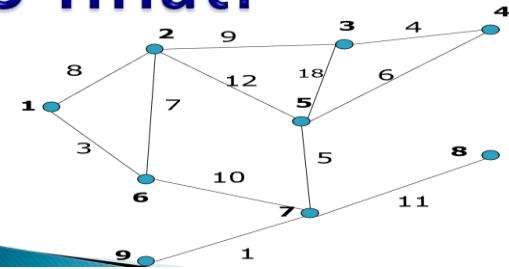
Thuật toán tìm cây bao trùm nhỏ nhất: Thuật toán KRUSKAL

- Bước 1: Liệt kê tất cả cạnh với trọng số của cạnh đó:**
 - Dựa vào đồ thị ta liệt kê ra các cạnh gồm đỉnh đầu, đỉnh cuối và trọng số:



Đỉnh đầu	Đỉnh cuối	Trọng số
1	2	8
1	6	3
2	3	9
2	5	12
2	6	7
3	4	4
3	5	18
4	5	6
5	7	5
6	7	10
7	8	11
7	9	1

Thuật toán tìm cây bao trùm nhỏ nhất: Thuật toán KRUSKAL



- **Bước 2: Sắp xếp các cạnh theo trọng số tăng dần**

Đỉnh đầu	Đỉnh cuối	Trọng số
7	9	1
1	6	3
3	4	4
5	7	5
4	5	6
2	6	7
1	2	8
2	3	9
6	7	10
7	8	11
2	5	12
3	5	18

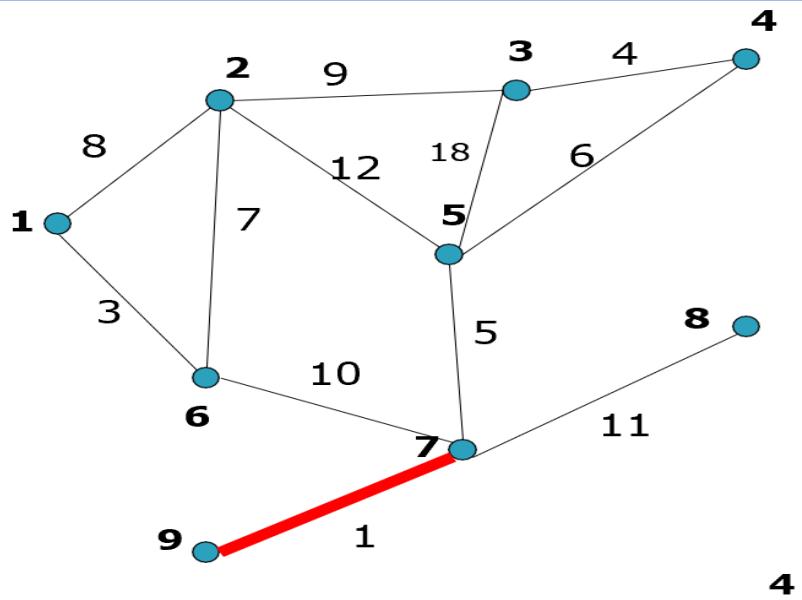
Thuật toán tìm cây bao trùm nhỏ nhất: Thuật toán KRUSKAL

- **Bước 3:** Dựa vào kết quả ở bước 2. Ta tiến hành tìm cây khung bằng thuật toán Kruskal. Dựa vào thứ tự bảng trên để đánh giá cạnh đó có thuộc cây khung tối thiểu hay không. Một cạnh thỏa điều kiện khi nó không góp phần tạo thành một chu trình với tất cả các cạnh đã tìm được.

Đỉnh đầu	Đỉnh cuối	Trọng số
7	9	1
1	6	3
3	4	4
5	7	5
4	5	6
2	6	7
1	2	8
2	3	9
6	7	10
7	8	11
2	5	12
3	5	18

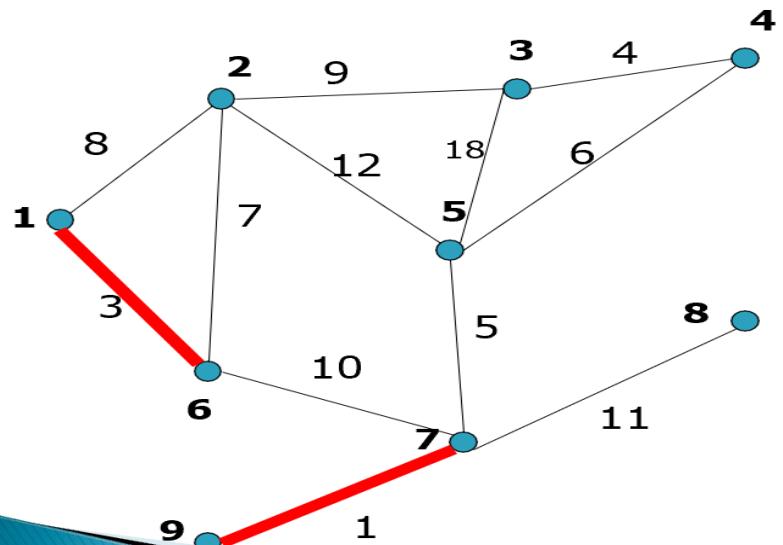
Thuật toán tìm cây bao trùm nhỏ nhất: Thuật toán KRUSKAL

KẾT QUẢ



CẠNH ĐANG XÉT

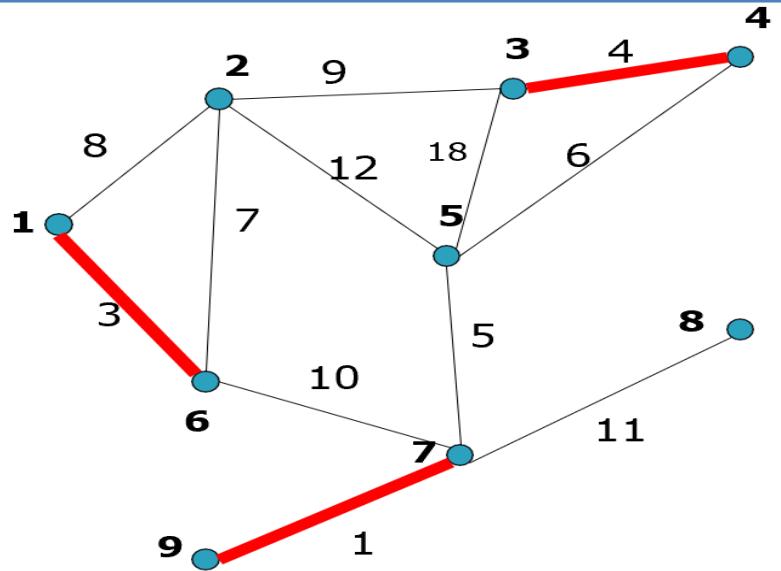
7-9-1: Ta nhận thấy cạnh 7-9 không tạo ra một chu trình nào. Vì vậy, thêm 7-9 vào tập hợp



1-6-3: Ta nhận thấy cạnh 1-6 không tạo ra một chu trình nào. Vì vậy, thêm 1-6 vào tập hợp

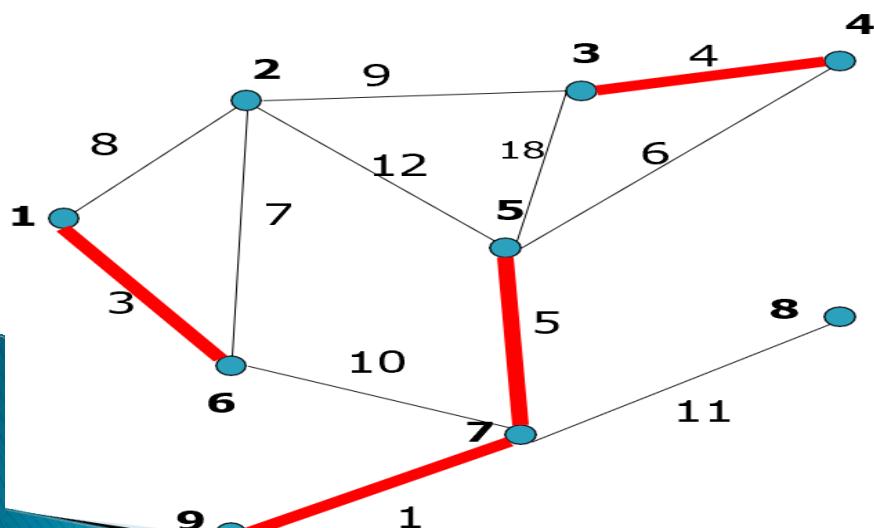
Thuật toán tìm cây bao trùm nhỏ nhất: Thuật toán KRUSKAL

KẾT QUẢ



CẠNH ĐANG XÉT

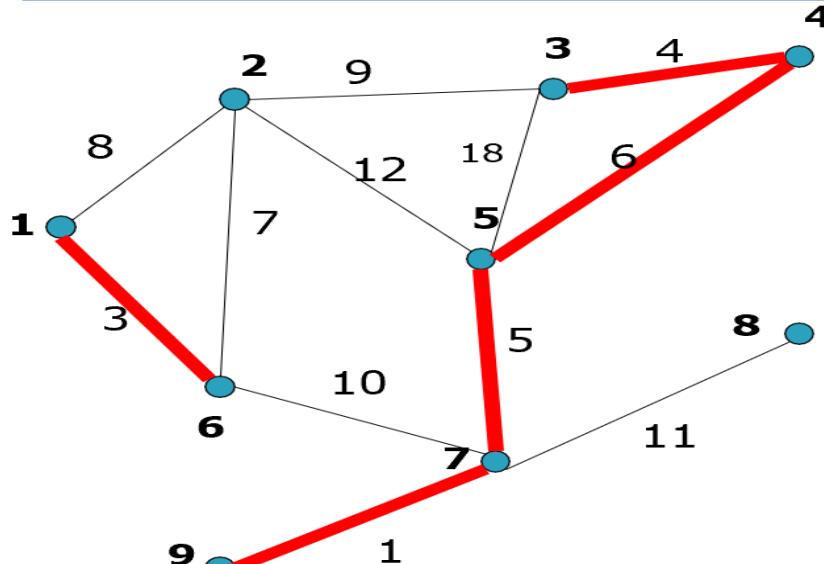
3-4-4: Ta nhận thấy cạnh 3-4 không tạo ra một chu trình nào. Vì vậy, thêm 3-4 vào tập hợp



5-7-5: Ta nhận thấy cạnh 5-7 không tạo ra một chu trình nào. Vì vậy, thêm 5-7 vào tập hợp

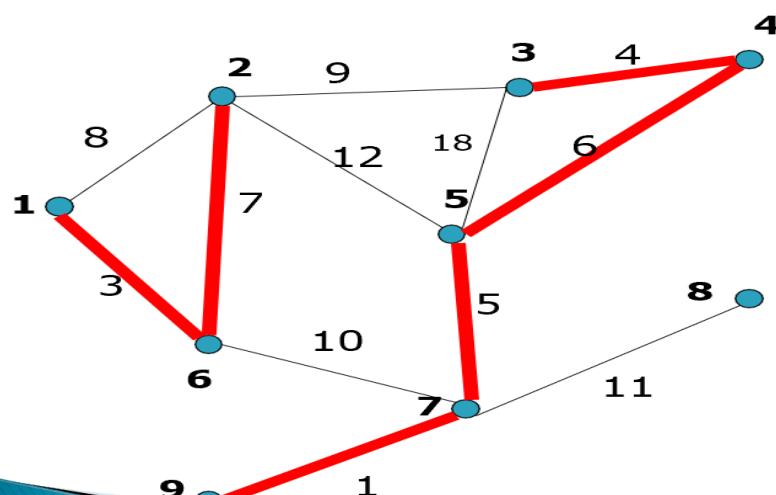
Thuật toán tìm cây bao trùm nhỏ nhất: Thuật toán KRUSKAL

KẾT QUẢ



CẠNH ĐANG XÉT

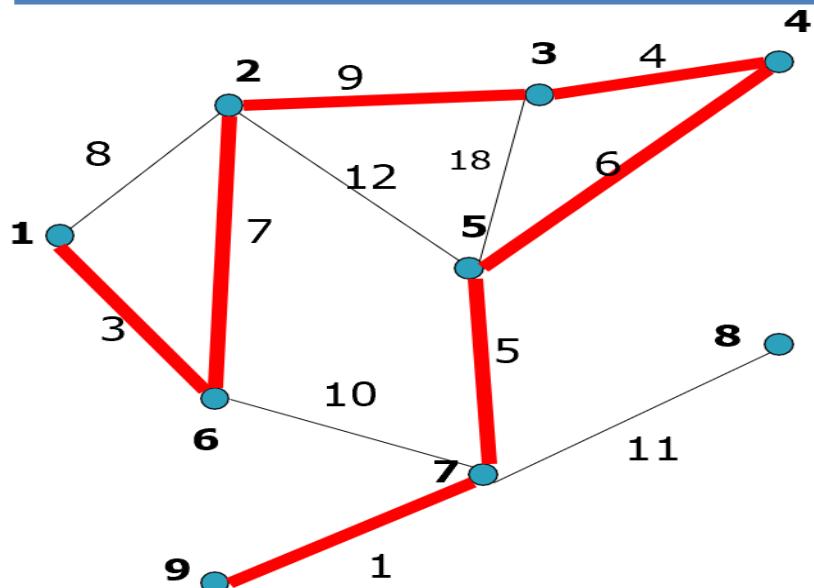
4-5-6: Ta nhận thấy cạnh 4-5 không tạo ra một chu trình nào. Vì vậy, thêm 4-5 vào tập hợp



2-6-7: Ta nhận thấy cạnh 2-6 không tạo ra một chu trình nào. Vì vậy, thêm 2-6 vào tập hợp

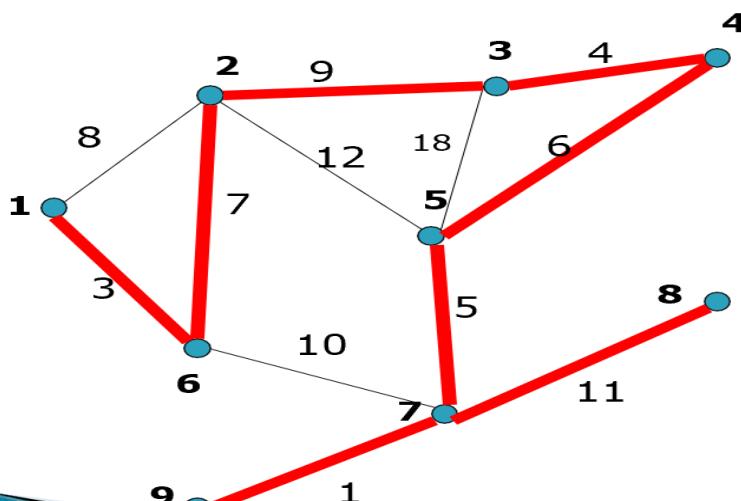
Thuật toán tìm cây bao trùm nhỏ nhất: Thuật toán KRUSKAL

KẾT QUẢ



CẠNH ĐANG XÉT

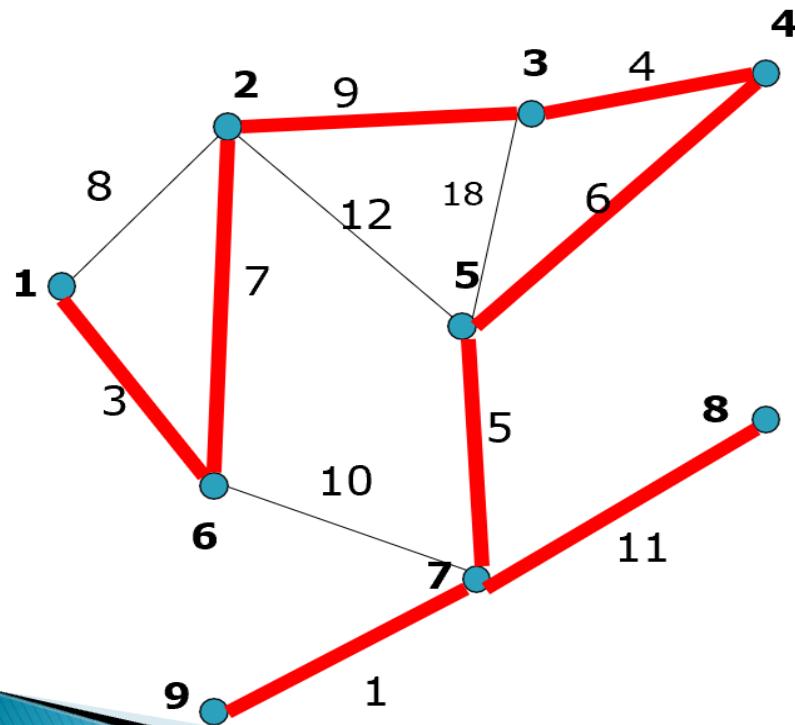
2-3-9: Ta nhận thấy cạnh 2-3 không tạo ra một chu trình nào. Vì vậy, thêm 2-3 vào tập hợp



7-8-11: Ta nhận thấy cạnh 7-8 không tạo ra một chu trình nào. Vì vậy, thêm 7-8 vào tập hợp

Thuật toán tìm cây bao trùm nhỏ nhất: Thuật toán KRUSKAL

- Đến đây, ta đã tìm được 8 cạnh. Vậy kết thúc thuật toán. (Thỏa (*))
- Kết quả: Ta được đồ thị sau

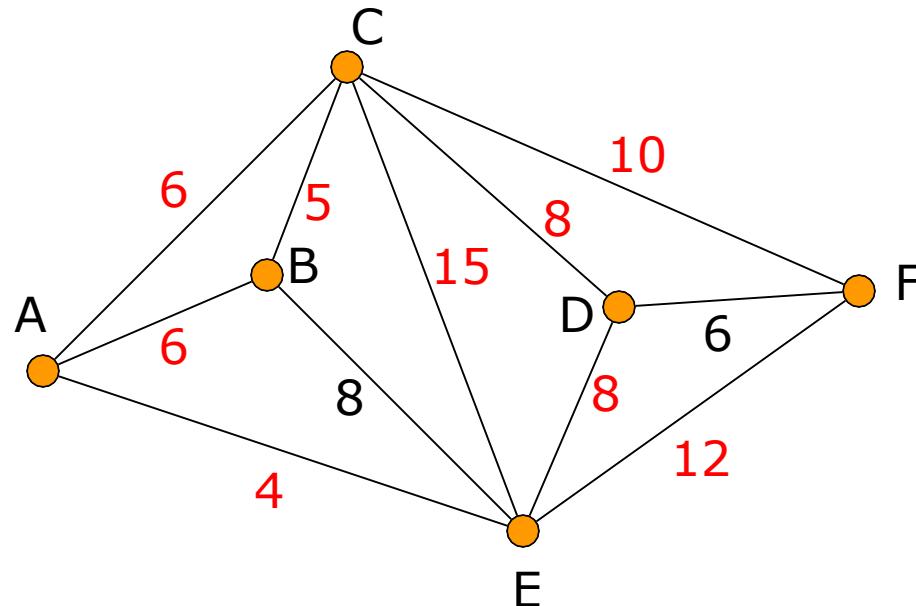


Với tổng chi phí là: Ta cộng tất cả các trọng số giữa các đỉnh lại với nhau

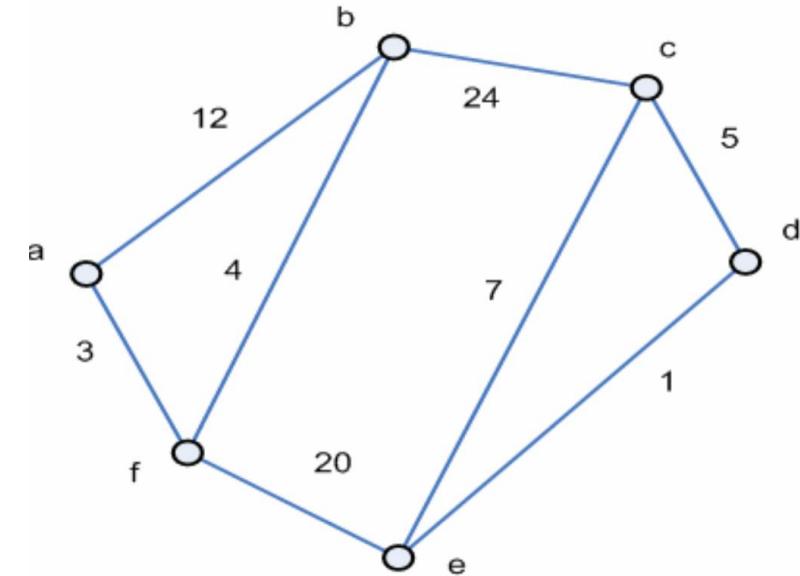
• Vậy tổng chi phí: $3 + 7 + 9 + 4 + 6 + 5 + 1 + 11 = 46$

Thuật toán tìm cây bao trùm nhỏ nhất: Thuật toán KRUSKAL

Sử dụng thuật toán Kruskal tìm một cây bao trùm nhỏ nhất của G, H? ghi ra kết quả từng bước chạy.



G



H

Thuật toán tìm cây bao trùm nhỏ nhất: Thuật toán PRIM

❖ Các bước của thuật toán Prim(G: liên thông, có trọng số, v_0 : đỉnh bắt đầu)

- **Bước 1:** Chọn một cạnh bất kỳ có trọng số nhỏ nhất từ đỉnh V_0 , đặt nó vào cây khung.
- **Bước 2:** Lần lượt ghép vào cây các cạnh có trọng số nhỏ nhất **liên thuộc** với một đỉnh của cây và không tạo ra chu trình trong cây.
- **Bước 3:** Thuật toán dừng lại khi $(n - 1)$ cạnh được ghép vào cây

Thuật toán tìm cây bao trùm nhỏ nhất: Thuật toán PRIM

Prim(G: liên thông, có trọng số, v_0 : đỉnh bắt đầu)

begin $E_T := \emptyset; V_T = \{v_0\}; // V_T là tập đỉnh của T, E_T: tập cạnh của T$

while ($|E_T| < n - 1$)

begin

 Tính $T(V_T) = \{(i,j) \in E / i \in V_T \wedge j \in V - V_T\}$

 Chọn cạnh $e = (u,v)$ trong $T(V_T)$ có trọng số bé nhất,
bổ sung e vào T (Nghĩa là: $E_T = E_T \cup \{e\}; V_T = V_T \cup \{v\}$)

end

return $T = \langle V_T, E_T \rangle;$

end

Thuật toán tìm cây bao trùm nhỏ nhất: Thuật toán PRIM

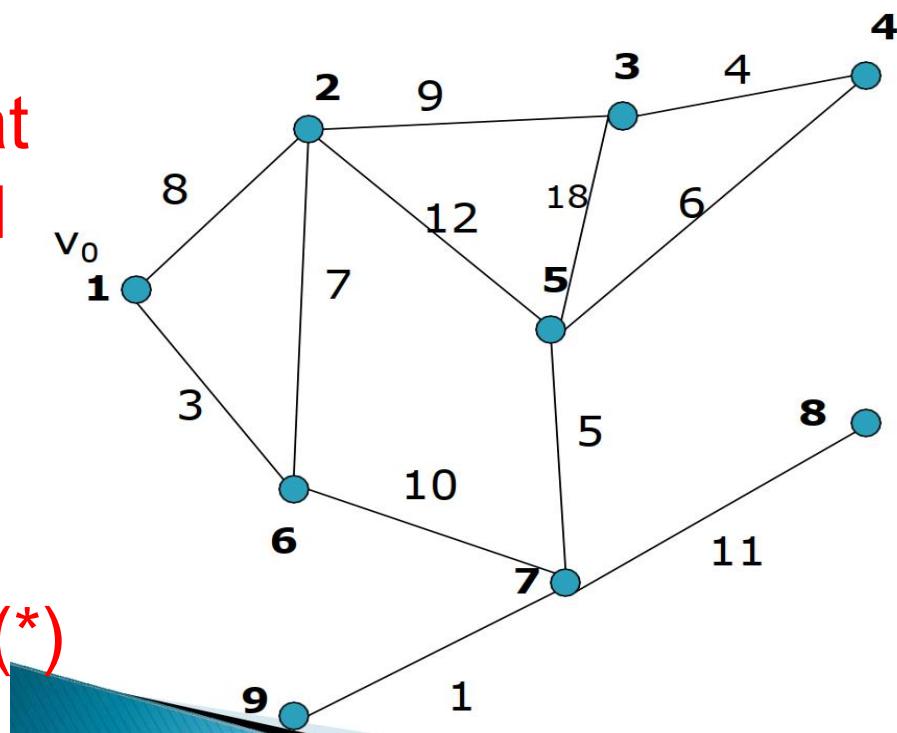
Ví dụ: Tìm cây bao trùm nhỏ nhất của đồ thị bên dưới bằng thuật toán Prim bắt đầu tại đỉnh 1, ghi ra kết quả từng bước chạy.

• G gồm có 9 đỉnh

• Đồ thị G có n phần tử. Thuật toán Prim sẽ dừng khi có $n-1$ trong tập hợp T

$$\bullet n = 9$$

$$\bullet \text{Vậy số cạnh trong tập hợp } T: n - 1 = 9 - 1 = 8(*)$$



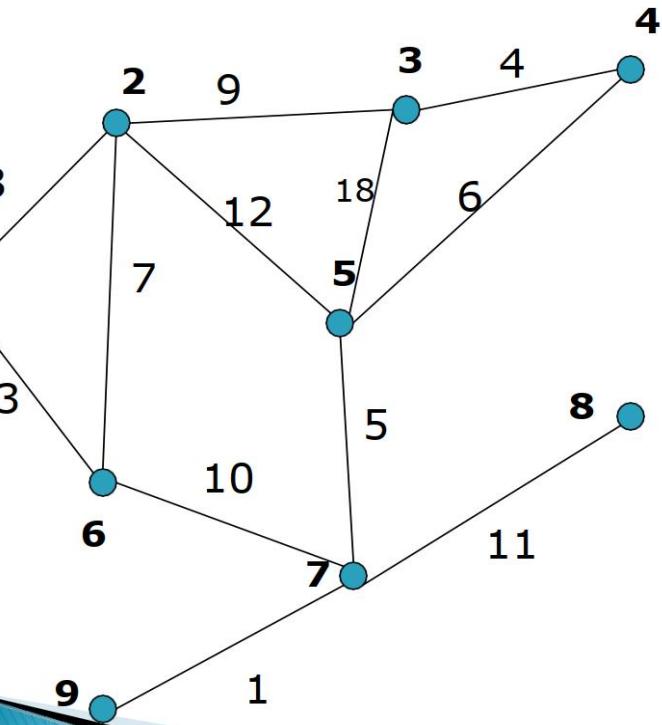
Thuật toán tìm cây bao trùm nhỏ nhất: Thuật toán PRIM

- Bước 1. Khởi tạo**

- Ta tạo ra 2 tập hợp V_T và E_T , V_T là tập hợp các đỉnh và E_T là tập hợp các cạnh tạo ra cây khung nhỏ nhất.
- Ban đầu $V_T = \{\emptyset\}$ (rỗng) và $E_T = \{\emptyset\}$.

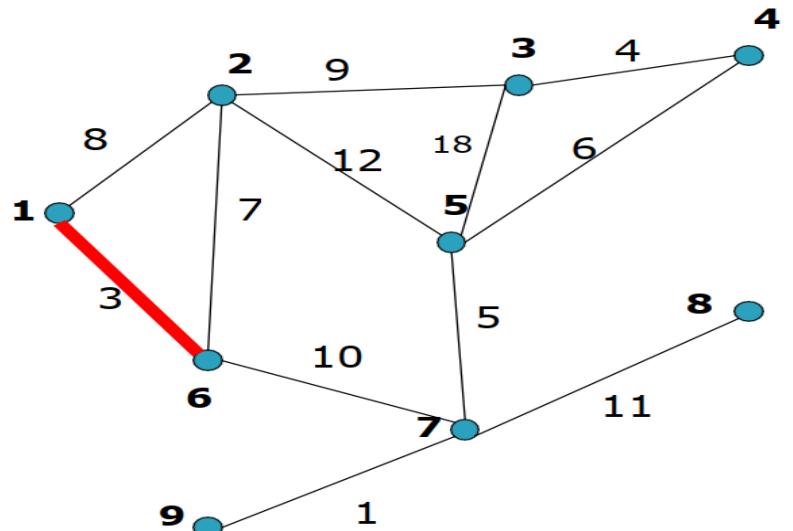
- Bước 2. Chọn 1 đỉnh ngẫu nhiên từ cây khung trên, ví dụ chúng ta chọn đỉnh theo yêu cầu bài toán là đỉnh 1.**

Sau bước này, ta có $V_T = \{1\}$ và $E_T = \{\emptyset\}$



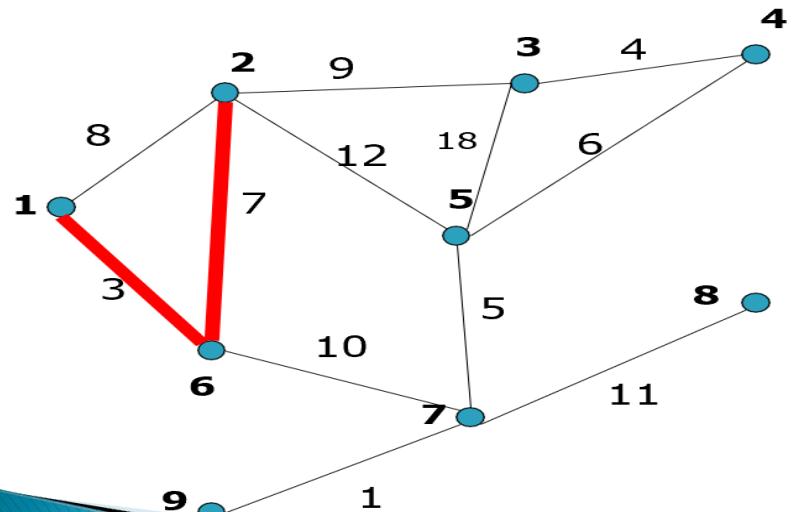
Thuật toán tìm cây bao trùm nhỏ nhất: Thuật toán PRIM

KẾT QUẢ



CẠNH ĐANG XÉT

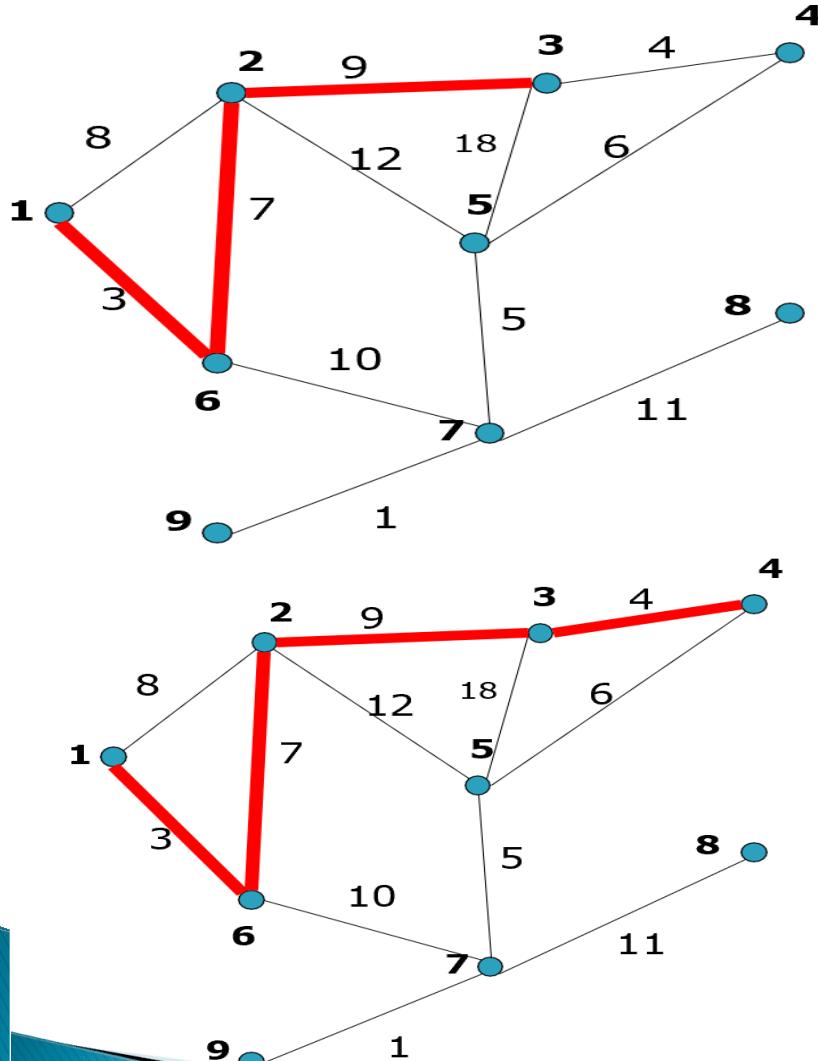
Tại đỉnh 1 trong V_T . Ở đây, ta sẽ tìm được 2 cạnh 1-2-8 và 1-6-3 do khoảng cách nhỏ nhất với giá trị là 3. Sau bước này ta có $V_T=\{1, 6\}$ và $E_T=\{(1, 6)\}$.



Duyệt tất cả các đỉnh trong V_T với các đỉnh ngoài V_T để tìm các cạnh kề còn lại chưa được xét, ta thấy cạnh 6-2-7 có trọng số nhỏ nhất và thêm nó vào trong cây đó là cạnh 6-2-7. Sau bước này ta có $V_T=\{1, 6, 2\}$ và $E_T=\{(1, 6), (6,2)\}$.

Thuật toán tìm cây bao trùm nhỏ nhất: Thuật toán PRIM

KẾT QUẢ



CẠNH ĐANG XÉT

Duyệt tất cả các đỉnh trong V_T với các đỉnh ngoài V_T để tìm các cạnh kề còn lại chưa được xét. Ở đây, ta sẽ tìm được cạnh 2-3-9 có trọng số nhỏ nhất với giá trị là 9. Sau bước này ta có $V_T=\{1, 6, 2, 3\}$ và $E_T=\{(1, 6), (6,2), (2,3)\}$.

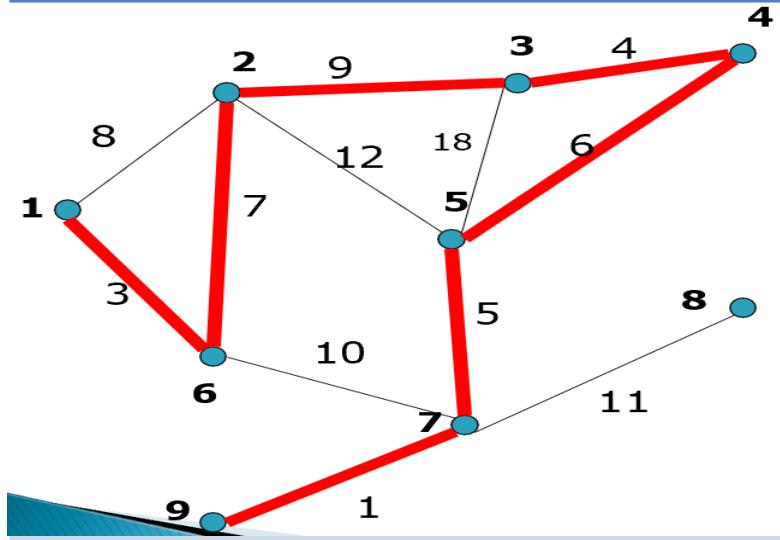
Duyệt tất cả các đỉnh trong V_T với các đỉnh ngoài V_T để tìm các cạnh kề còn lại chưa được xét. Ở đây, ta sẽ tìm được cạnh 3-4-4 do trọng số nhỏ nhất với giá trị là 4. Sau bước này ta có $V_T=\{1, 6, 2, 3, 4\}$ và $E_T=\{(1, 6), (6,2), (2,3),(3,4)\}$.

Thuật toán tìm cây bao trùm nhỏ nhất: Thuật toán PRIM

KẾT QUẢ	CẠNH ĐANG XÉT
	Duyệt tất cả các đỉnh trong V_T với các đỉnh ngoài V_T để tìm các cạnh kề còn lại chưa được xét. Ở đây, ta sẽ tìm được cạnh 4-5-6 có trọng số nhỏ nhất với giá trị là 6. Sau bước này ta có $V_T=\{1, 6, 2, 3, 4, 5\}$ và $E_T=\{(1, 6), (6,2), (2.3), (3,4), (4,5)\}$.
	Duyệt tất cả các đỉnh trong V_T với các đỉnh ngoài V_T để tìm các cạnh kề còn lại chưa được xét. Ở đây, ta sẽ tìm được cạnh 5-7-5 có trọng số nhỏ nhất với giá trị là 5. Sau bước này ta có $V_T=\{1, 6, 2, 3, 4, 5, 7\}$ và $E_T=\{(1, 6), (6,2), (2.3), (3,4), (4,5), (5,7)\}$.

Thuật toán tìm cây bao trùm nhỏ nhất: Thuật toán PRIM

KẾT QUẢ



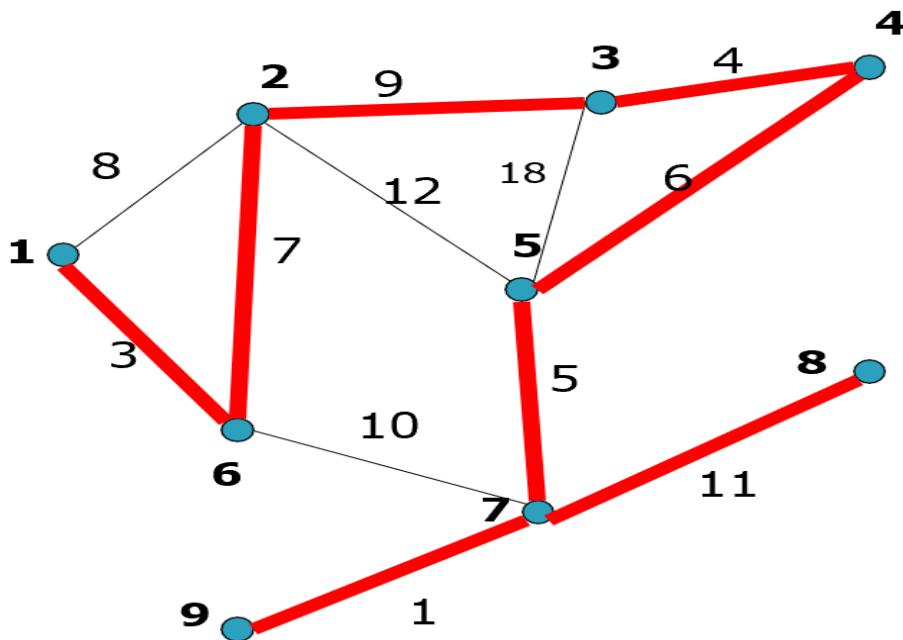
CẠNH ĐANG XÉT

Duyệt tất cả các đỉnh trong V_T với các đỉnh ngoài V_T để tìm các cạnh kề còn lại chưa được xét. Ở đây, ta sẽ tìm được cạnh 7-9-1, có trọng số nhỏ nhất với giá trị là 1. Sau bước này ta có $V_T=\{1, 6, 2, 3, 4, 5, 7, 9\}$ và $E_T=\{(1, 6), (6,2), (2,3), (3,4), (4,5), (5,7), (7,9)\}$.

Duyệt tất cả các đỉnh trong V_T với các đỉnh ngoài V_T để tìm các cạnh kề còn lại chưa được xét. Ở đây, ta sẽ tìm được cạnh 7-8-11 có trọng số nhỏ nhất giá trị là 5. Sau bước này ta có $V_T=\{1, 6, 2, 3, 4, 5, 7, 9, 8\}$ và $E_T=\{(1, 6), (6,2), (2,3), (3,4), (4,5), (5,7), (7,9), (7,8)\}$.

Thuật toán tìm cây bao trùm nhỏ nhất: Thuật toán PRIM

- Đến đây, ta đã tìm được 8 cạnh. Vậy kết thúc thuật toán. (Thỏa (*))
- Kết quả: Ta được đồ thị sau

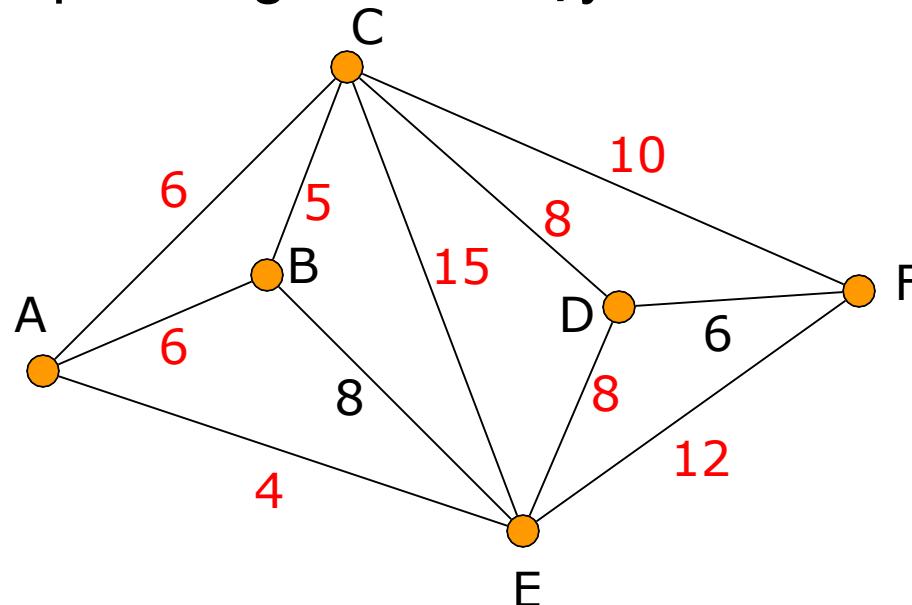


Với tổng chi phí là: Ta cộng tất cả các trọng số giữa các đỉnh lại với nhau

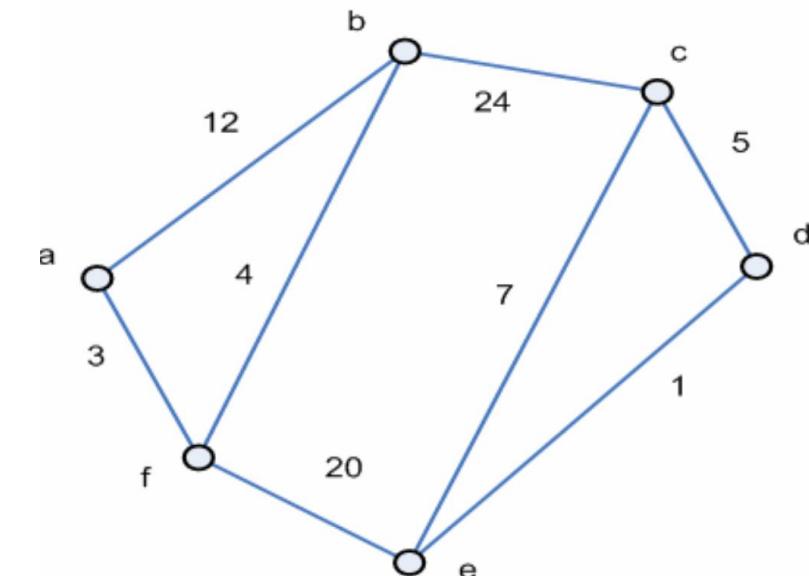
• Vậy tổng chi phí: $3 + 7 + 9 + 4 + 6 + 5 + 1 + 11 = 46$

Thuật toán tìm cây bao trùm nhỏ nhất: Thuật toán PRIM

Sử dụng thuật toán Prim tìm một cây bao trùm nhỏ nhất của G, H bắt đầu từ đỉnh A? ghi ra kết quả từng bước chạy.



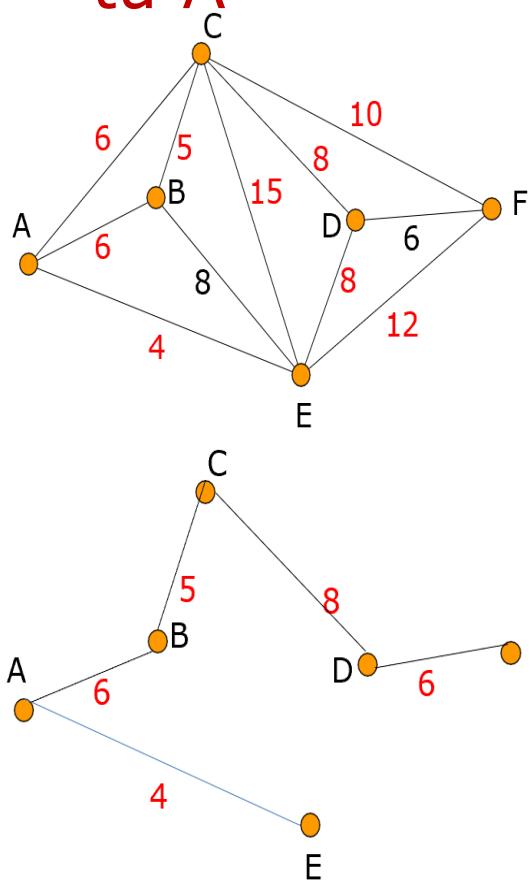
G



H

Thuật toán tìm cây bao trùm nhỏ nhất: Thuật toán PRIM

Lời giải với đồ thị G, dùng thuật toán Prim bắt đầu từ A

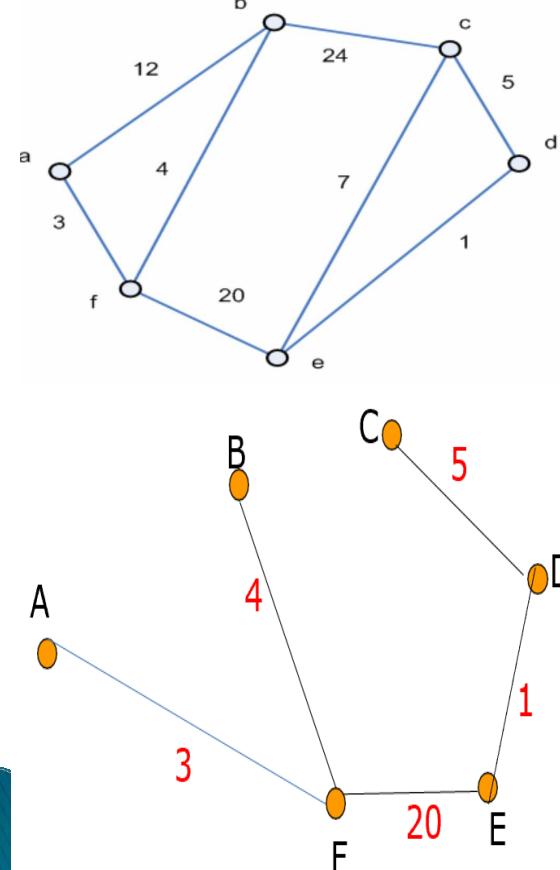


Bước	Tập đỉnh trong cây T	Tập cạnh Cây T (Xét tất cả các đỉnh ngoài cây T tạo cạnh kề với Tập đỉnh trong T tại mỗi bước và chọn cạnh có trọng số min)
1	{A}	(A, E) - 4
2	{A,E}	Bổ sung thêm (A, B)- 6
2	{A,E,B}	Bổ sung thêm (B, C) - 5
3	{A,E,B,C}	Bổ sung thêm (C, D) - 8
4	{A,E,B,C,D}	Bổ sung thêm (B, F) -6
5	{A,E,B,C,D,F}	Kết thúc vì các đỉnh đã duyệt xong

Cây bao trùm có trọng số nhỏ nhất là $w(T) = 4+6+5+8+6 = 29$

Thuật toán tìm cây bao trùm nhỏ nhất: Thuật toán PRIM

Lời giải với đồ thị H, dùng thuật toán Prim bắt đầu từ A



Bước	Tập đỉnh trong cây T	Tập cạnh Cây T (Xét tất cả các đỉnh ngoài cây T tạo cạnh kề với Tập đỉnh trong T tại mỗi bước và chọn cạnh có trọng số min)
1	{A}	(A, F) - 3
2	{A,F}	Bổ sung thêm (B, F)- 4
2	{A,F,B}	Bổ sung thêm (F, E) - 20
3	{A,F,B,E}	Bổ sung thêm (E, D) - 1
4	{A,F,B,E,D}	Bổ sung thêm (D, C) -5
5	{A,F,B,E,D,C}	Kết thúc vì các đỉnh đã duyệt xong

Cây bao trùm có trọng số nhỏ nhất là $w(T) = 3+4+20+1+5 = 33$