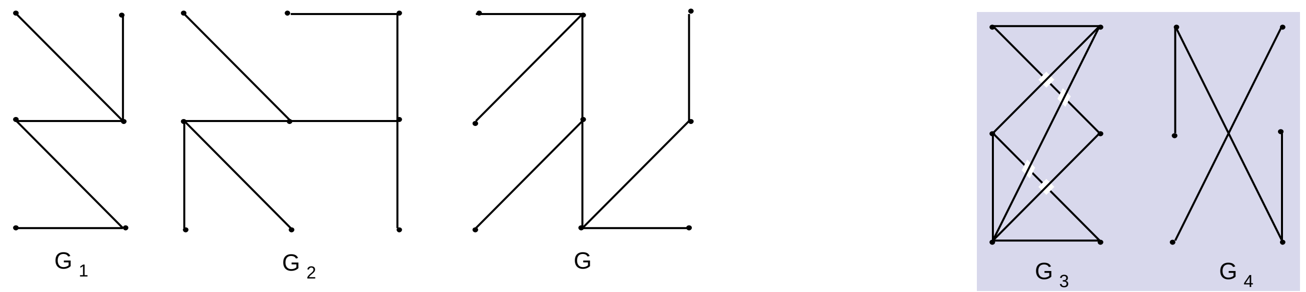
CHƯƠNG 6: CÂY

- **Một số khái niệm cơ bản**
- **Cây m – phân và các tính chất**
- Phép duyệt cây nhị phân
- Ký pháp nghịch đảo Ba Lan
- **Thuật toán Prim và Kruskal tìm cây khung nhỏ nhất trong đồ thị liên thông có trọng số**

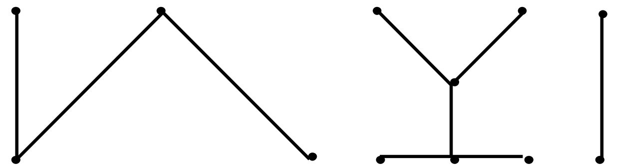
Một số khái niệm cơ bản

- Cây
  - Định nghĩa:
    - Cây là một đồ thị *vô hướng*, *liên thông* và *không có chu trình sơ cấp*
      - Cây không có cạnh bội và khuyên
      - Cây là một đơn đồ thị
  - Ví dụ



Một số khái niệm cơ bản

- Rừng
  - Định nghĩa:
    - Rừng là một đồ thị *vô hướng* và *không có chu trình*
      - Rừng có thể có nhiều thành phần liên thông
      - Mỗi thành phần liên thông là một cây
  - Ví dụ



G

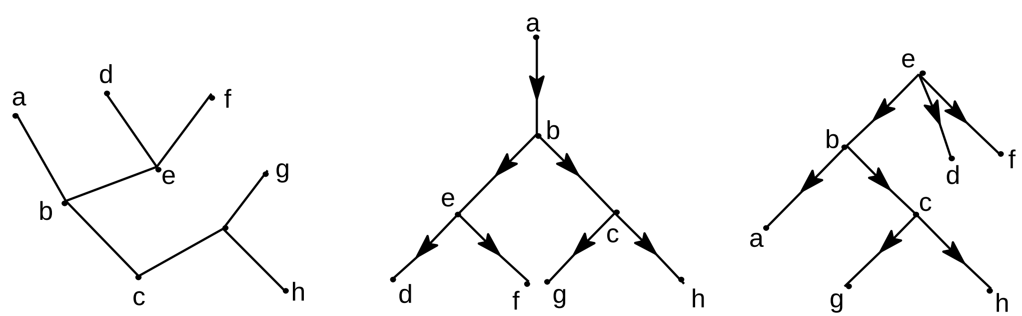
Một số khái niệm cơ bản

- Định lý (Điều kiện đủ của cây)
  - Nếu mọi cặp đỉnh của một đồ thị vô hướng G luôn tồn tại một đường đi sơ cấp duy nhất thì G là một cây.

  (Chứng minh SV tham khảo tài liệu)

Một số khái niệm cơ bản

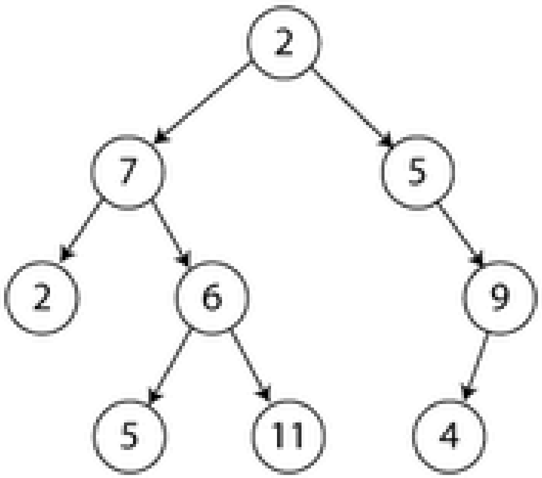
- Cây có gốc
  - Định nghĩa
    - Một cây với một đỉnh được chọn làm gốc
    - Định hướng các cạnh trên cây từ gốc đi ra
  - Ví dụ



- Cùng một cây, nếu chọn gốc khác nhau thì cây có gốc thu được sẽ khác nhau

Một số khái niệm cơ bản

- Cây có gốc
  - Một số khái niệm
    - Cha
    - Anh em
    - Tổ tiên
    - Con cháu
    - Lá
    - Đỉnh trong
    - Cây con
    - Mức
    - Chiều cao



Một số khái niệm cơ bản

- Định lý Daisy Chain

T là đồ thị có n đỉnh. Các mệnh đề tương đương:

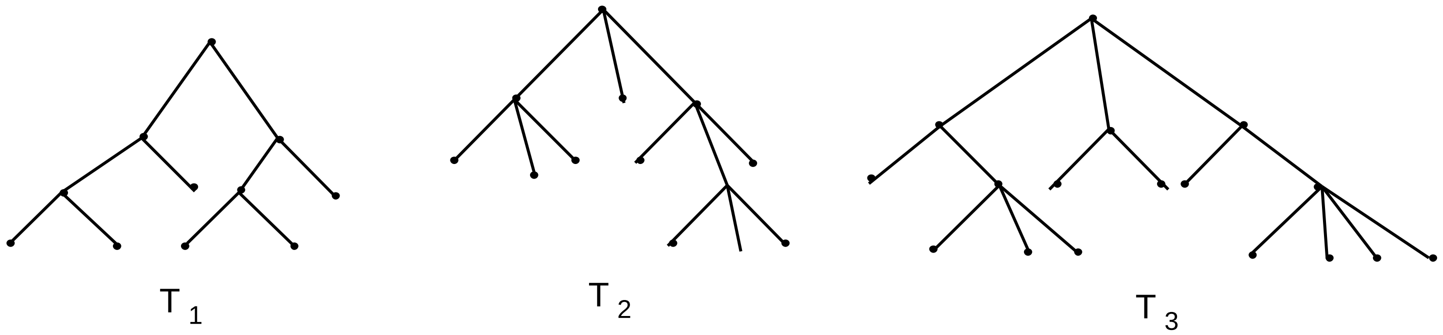
1. T là một cây
2. T không có chu trình và có *n*-1 cạnh
3. T liên thông, mọi cạnh đều là cầu
4. Giữa hai đỉnh bất kỳ của T luôn tồn tại một đường đi sơ cấp duy nhất
5. T không có chu trình và nếu thêm một cạnh mới nối 2 đỉnh bất kỳ của T thì sẽ tao ra một chu trình
6. T liên thông và có *n*-1 cạnh

Một số khái niệm cơ bản

- Cây *m*-phân
  - Định nghĩa
    - *Cây m-phân*
      - Cây có gốc
      - Tất cả các đỉnh trong có không quá *m* con
    - Cây *m*-phân đầy đủ
      - Cây có gốc
      - Tất cả các đỉnh trong có đúng *m* con
    - *m*=2: Cây nhị phân

Một số khái niệm cơ bản

- Cây *m*-phân
  - Ví dụ



- T_1: Cây nhị phân đầy đủ
- T_2: Cây tam phân đầy đủ
- T_3: Cây tứ phân (không đầy đủ)

Một số tính chất của cây

- **Tính chất 1:**
  - Cây n đỉnh ($n \geq 2$) có ít nhất 2 đỉnh treo
- **Tính chất 2:**
  - Cây m-phân đầy đủ với i đỉnh trong có

$$n = m.i + 1 \text{ đỉnh}$$

- **Tính chất 3:** Cho cây m-phân đầy đủ có n đỉnh, có i đỉnh trong và l lá. Khi đó:
  - $i = (n - 1)/m$
  - $l = [(m - 1)n + 1] / m$
  - $l = (m - 1)i + 1$
  - $n = l + i$

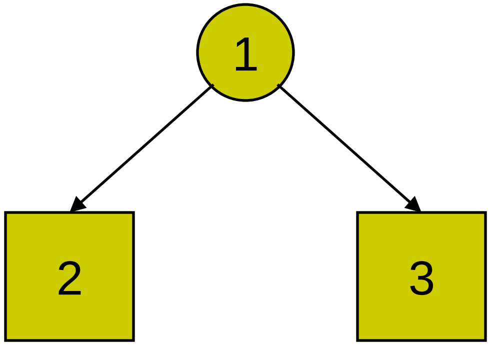
Phép duyệt cây nhị phân

- Định nghĩa
  - Duyệt cây
    - Liệt kê tất cả các đỉnh của cây theo một thứ tự xác định, mỗi đỉnh một lần
  - 3 phương pháp duyệt cây
    - Duyệt tiền tự (Pre-Oder)
    - Duyệt trung tự (In-Oder)
    - Duyệt hậu tự (Post-Oder)

Cả 3 phương pháp duyệt trên đều được định nghĩa đệ quy đối với cây nhị phân (mỗi nút có tối đa 2 con lần lượt được gọi là con trái và con phải của nút)

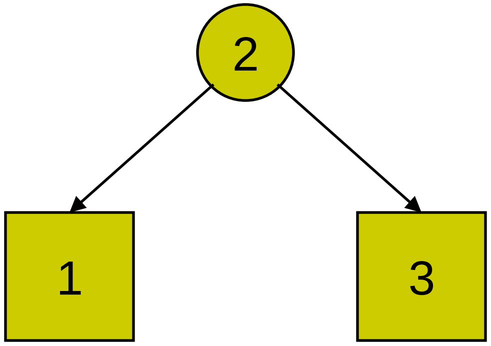
Phép duyệt cây nhị phân

- Định nghĩa
  - Duyệt tiền tự
    1. Duyệt nút gốc
    2. Duyệt tiền tự con trái
    3. Duyệt tiền tự con phải



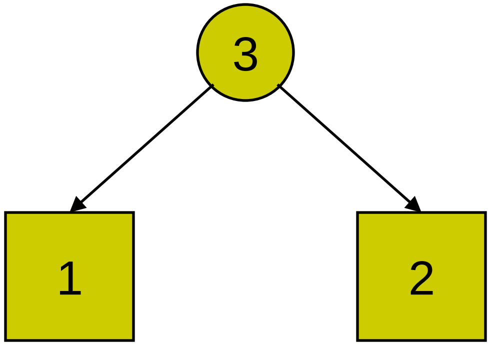
Phép duyệt cây nhị phân

- Định nghĩa
  - Duyệt trung tự
    1. Duyệt trung tự con trái
    2. Duyệt nút gốc
    3. Duyệt trung tự con phải



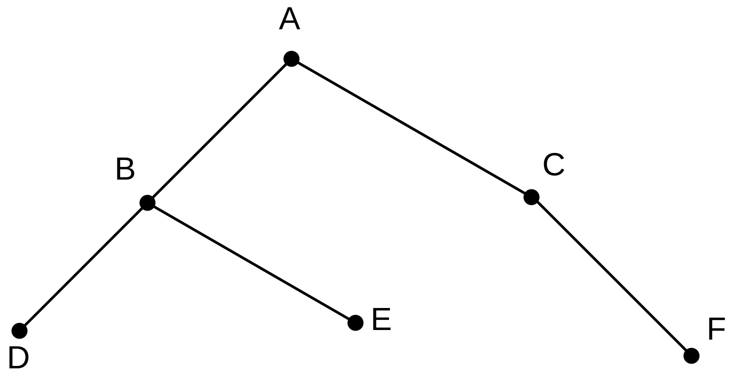
Phép duyệt cây nhị phân

- Định nghĩa
  - Duyệt hậu tự
    1. Duyệt hậu tự con trái
    2. Duyệt hậu tự con phải
    3. Duyệt nút gốc



Phép duyệt cây nhị phân

- Định nghĩa
  - Ví dụ
    - Duyệt tiền tự
      - A B D E C F
    - Duyệt trung tự
      - D B E A C F
    - Duyệt hậu tự
      - D E B F C A



Ký pháp nghịch đảo Ba Lan

- Cây biểu thức số học
  - Là cây nhị phân
    - Mỗi nút trong biểu diễn cho một toán tử 2 ngôi θ
    - Mỗi nút lá biểu diễn cho một toán hạng của biểu thức

Nếu nút trong biểu diễn cho toán tử 2 ngôi θ và có 2 con:

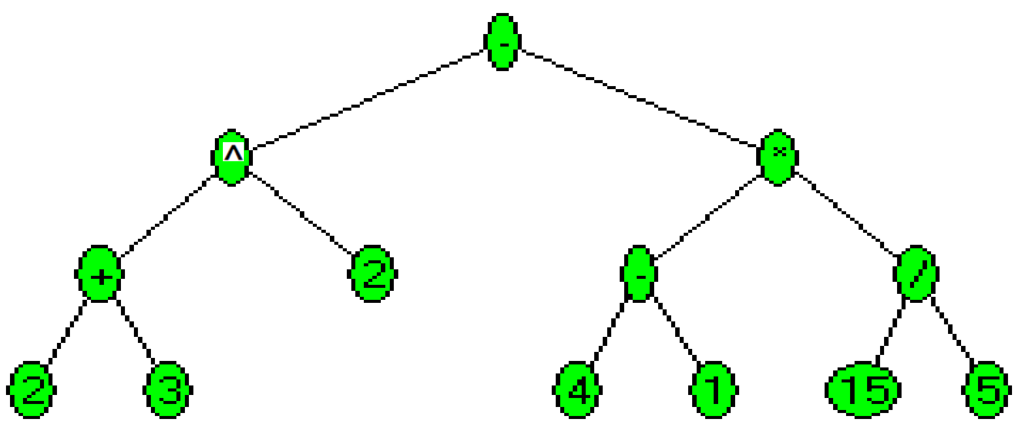
- Con trái biểu diễn cho biểu thức E_1
- Con phải biểu diễn cho biểu thức E_2

khi đó nút trong này biểu diễn cho biểu thức $E_1 \, \theta \, E_2$

Ký pháp nghịch đảo Ba Lan

- Cây biểu thức số học
  - Ví dụ:

$$E = (2 + 3)^2 – (4 – 1)*(15/5)$$



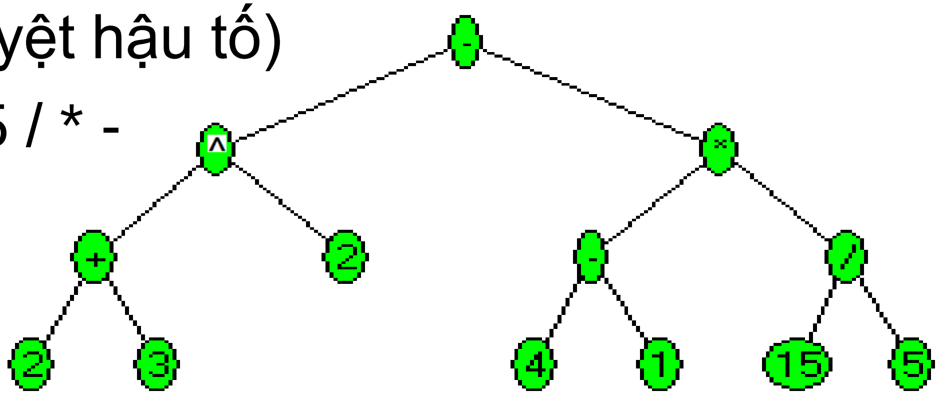
Ký pháp nghịch đảo Ba Lan

- Cây biểu thức số học
  - Duyệt cây biểu thức
    - Biểu thức tiền tố (duyệt tiền tự)

      \- ^ + 2 3 2 * - 4 1 / 15 5
    - Biểu thức trung tố (duyệt trung tố)

      2 + 3 ^ 2 - 4 - 1 * 15 / 5
    - Biểu thức hậu tố (duyệt hậu tố)

      2 3 + 2 ^ 4 1 - 15 5 / * -



Ký pháp nghịch đảo Ba Lan

- Cây biểu thức số học
  - Ký pháp nghịch đảo Ba Lan (**R**everse **P**olish **N**otation – RPN)
    - Biểu thức ở dạng hậu tố
    - Sử dụng để tính giá trị biểu thức trên máy tính
      - Tính từ trái qua phải
      - Không sử dụng dấu ngoặc
      - Sử dụng Stack (ngăn xếp)

Ký pháp nghịch đảo Ba Lan

- Cây biểu thức số học
  - Ký pháp nghịch đảo Ba Lan (**R**everse **P**olish **N**otation – RPN)
    - Thuật toán tính giá trị biểu thức RPN
      - Đọc một ký hiệu (token)
      - Nếu ký hiệu là một số
        - Đẩy vào Stack
      - Ngược lại, ký hiệu là một toán tử
        - Lấy ra 2 toán hạng từ Stack
        - Tính giá trị theo toán tử đối với 2 toán hạng
        - Đẩy kết quả vào Stack

Ký pháp nghịch đảo Ba Lan

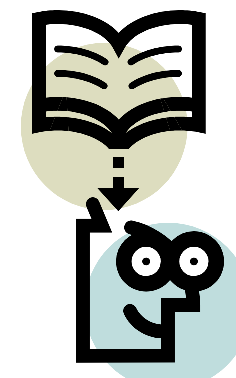
- Cây biểu thức số học
  - Ký pháp nghịch đảo Ba Lan (**R**everse **P**olish **N**otation – RPN)

  Ví dụ: Tính giá trị biểu thức E = (2 + 3)^2 – (4 1)*(15/5)

  - Biểu thức nhập dưới dạng ký pháp RPN

  - E = 2 3 + 2 ^ 4 1 - 15 5 / * -

  - Quá trình lưu trữ của cấu trúc Stack như sau:

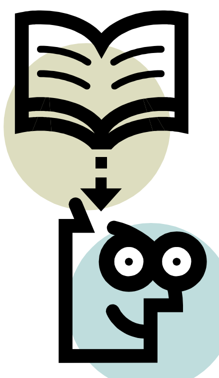


Ký pháp nghịch đảo Ba Lan

Ví dụ: E = 2 3 + 2 ^ 4 1 - 15 5 / * -

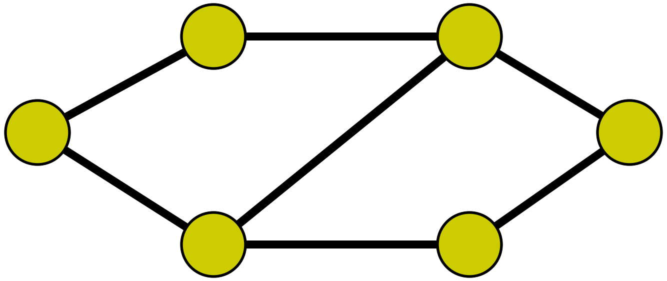
- Quá trình lưu trữ của cấu trúc Stack như sau:

16
-
9
*
3
/
5
15
3
-
1
4
25
^
2
5
+
3
2



Cây khung (Spanning Tree)

- Định nghĩa
  - Cây khung của đơn đồ thị G
    - Đồ thị con của G
    - Chứa tất cả các đỉnh của G
  - Một đồ thị có thể có nhiều cây khung
- Ví dụ



Cây khung (Spanning Tree)

- Định lý
  - Một đơn đồ thị là liên thông khi và chỉ khi nó có cây khung

(Chứng minh xem tài liệu)

Cây khung (Spanning Tree)

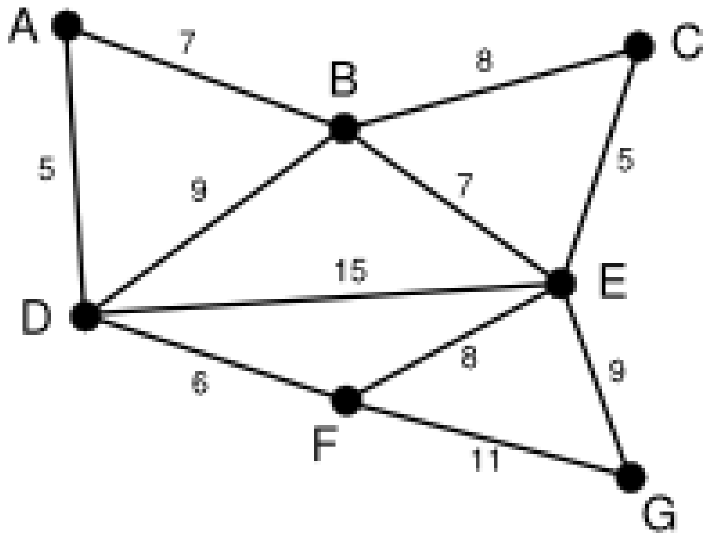
- **Cây khung nhỏ nhất**
  - **Định nghĩa**
    - Cây khung nhỏ nhất trong một đồ thị liên thông, có trọng số là một cây khung có tổng trọng số trên các cạnh của nó là nhỏ nhất.

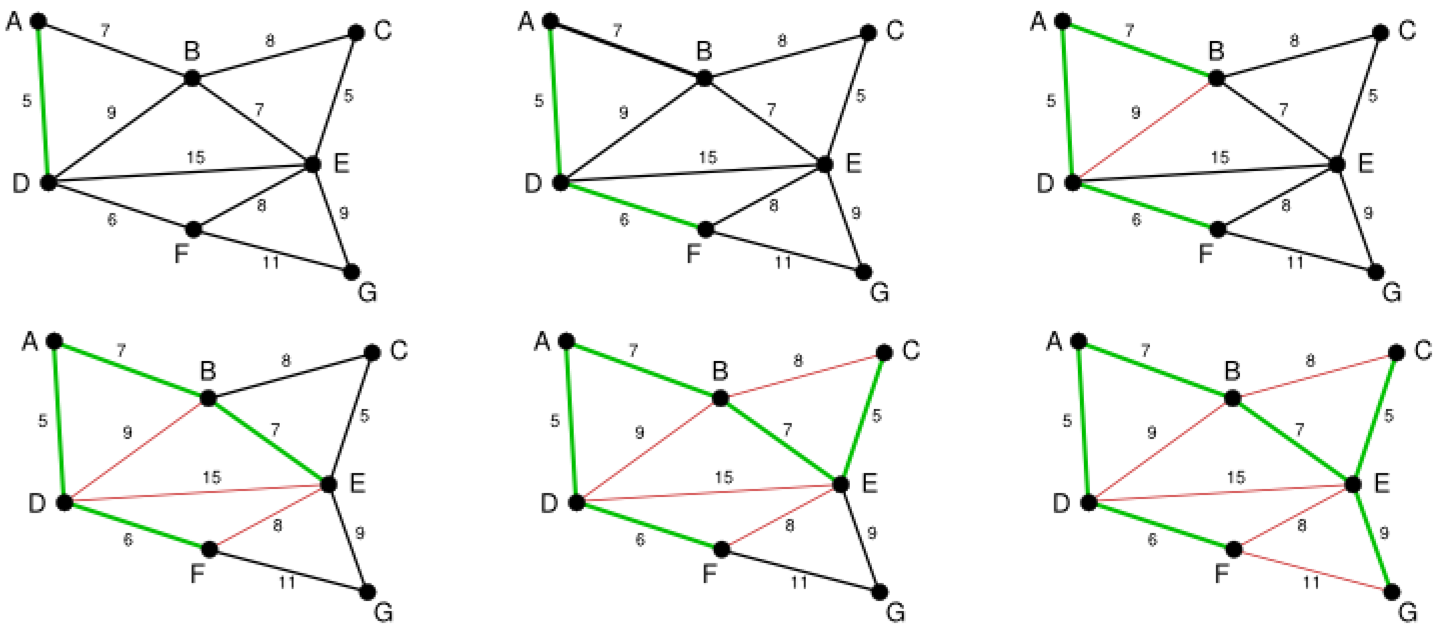
Cây khung (Spanning Tree)

- Cây khung nhỏ nhất
  - Thuật toán Prim
    - Bắt đầu bằng việc chọn một đỉnh bất kỳ, đặt nó vào cây khung T.
    - Trong khi cây khung T có ít hơn n đỉnh
      - Ghép vào T cạnh có trọng số nhỏ nhất liên thuộc với một đỉnh của T và không tạo ra chu trình trong T.

Chú ý: - Thuật toán dừng lại khi Tcó đủ n đỉnh hay (n-1) cạnh.

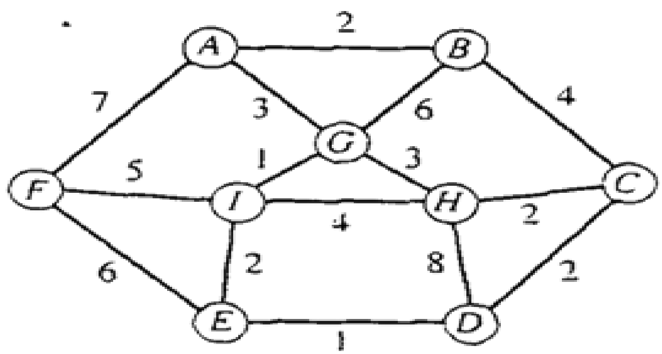
- Có nhiều hơn một cây khung nhỏ nhất ứng với một đồ thị liên thông có trọng số.


A
B
C
D
E
F
G
7
8
5
9
7
5
15
6
8
9
11

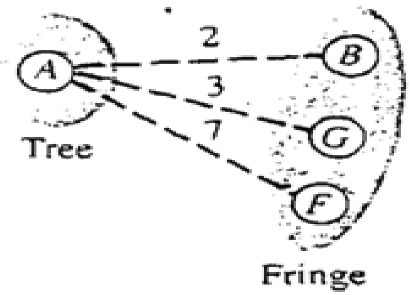


Cây khung (Spanning Tree)

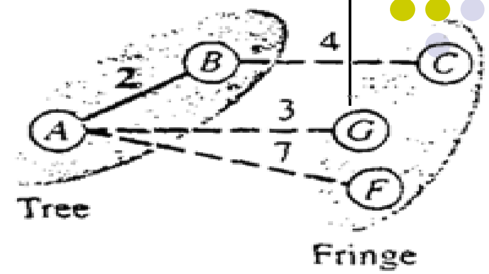
- Cây khung nhỏ nhất
  - Thuật toán Prim
    - Bước 1: Khởi tạo
      - $V_T = \{s\}$; $E_T = \varnothing$; (V_T – tập đỉnh; E_T – tập cạnh)
      - $d_s = 0$; $v \notin V_T$ $d_v = w(s, v)$, nếu s và v liền kề
      - $d_v = \infty$, nếu s và v không liền kề
    - Bước 2: Tìm cạnh
      - Tìm u mà $d_u = \min \{d_v \mid v \notin V_T\}$
      - $V_T = V_T \cup \{u\}$;
      - $E_T = E_T \cup \{e\}$, e – cạnh nối u với một đỉnh của cây có trọng số d_u
      - Nếu $V_T \equiv V$ thì dừng.
    - Bước 3: Cập nhật nhãn
      - $d_v = \min \{d_v, w(u, v)\}$ với $v \notin V_T$



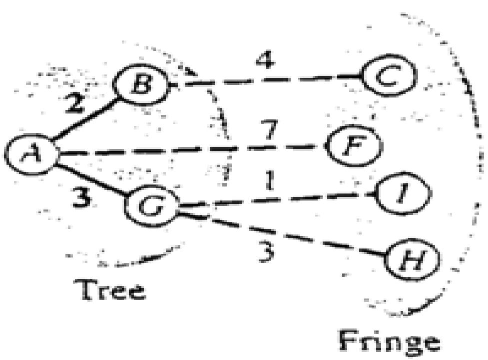
(a) A weighted graph



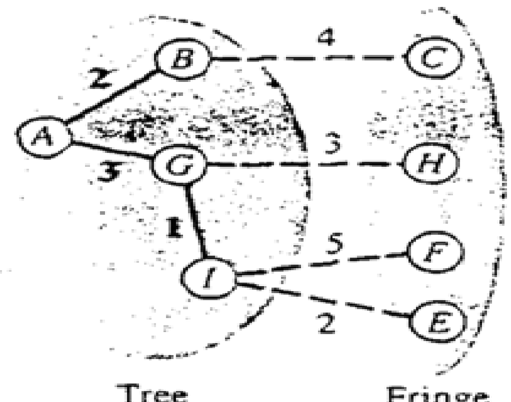
(b) After selection of the starting vertex



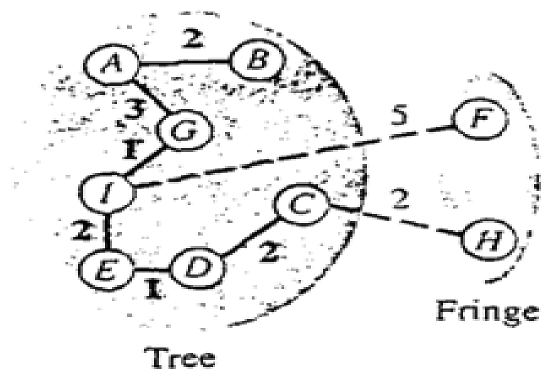
(c) BG was considered but did not replace AG as a candidate.



(d) After AG is selected and fringe and candidates are updated



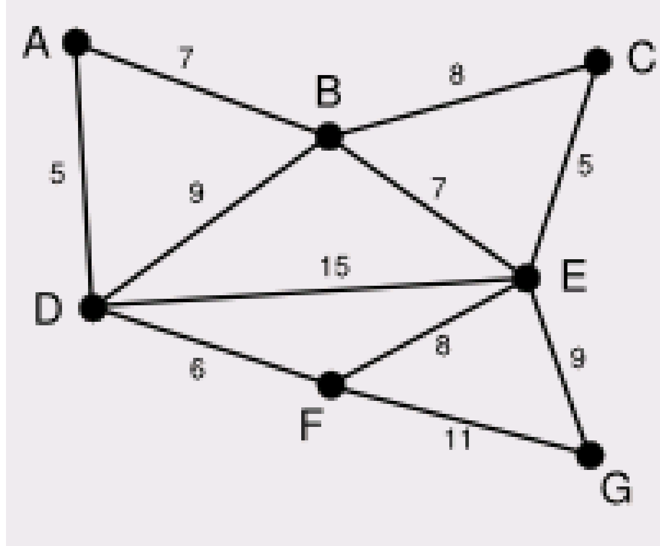
(e) IF has replaced AF as a candidate.

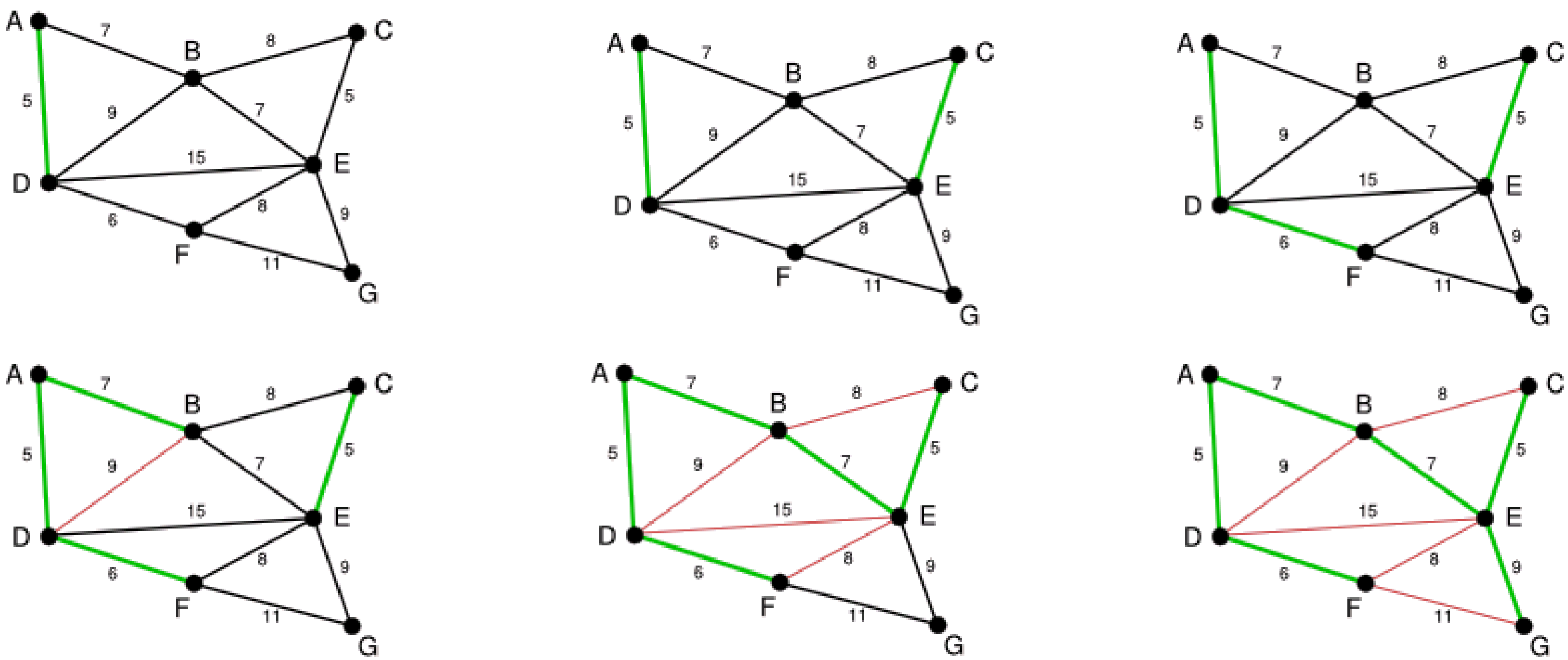


(f) After several more passes: The two candidate edges will be put in the tree.

Cây khung (Spanning Tree)

- Cây khung nhỏ nhất
  - Thuật toán Kruskal
    - Bắt đầu bằng việc chọn một cạnh có trọng số nhỏ nhất, đặt nó vào cây khung T.
    - Trong khi cây khung T có ít hơn (n-1) cạnh
      - Ghép vào T cạnh có trọng số nhỏ nhất và không tạo ra chu trình trong T.


A
B
C
D
E
F
G
7
8
5
9
7
5
15
8
9
6
11



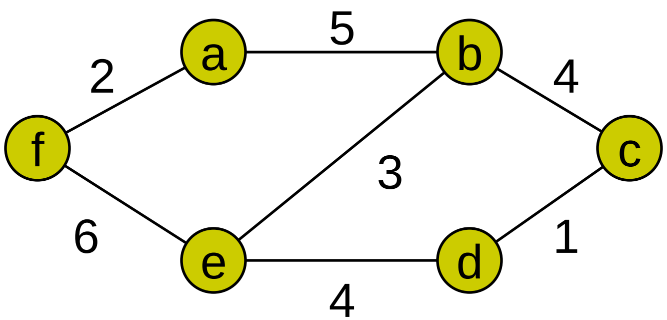
Cây khung (Spanning Tree)

- Cây khung nhỏ nhất
  - Thuật toán Kruskal
    - Bước 1:
      - Sắp xếp các cạnh của đồ thị G theo thứ tự có trọng số không giảm: $w(e_1) \leq w(e_2) \leq \ldots \leq w(e_m)$
      - $E_T = \{e_1\}$, i =1
    - Bước 2: Tìm $k = \min \{ j \mid E_T \cup \{e_j\}$ không có chu trình}

      $E_T = E_T \cup \{e_k\}$
    - Bước 3: i = i +1
      - Nếu i = n-1 thì dừng
      - Nếu i < n-1 thì quay lại bước 2

Cây khung (Spanning Tree)

- Cây khung nhỏ nhất
  - **Ví dụ:** Tìm cây khung nhỏ nhất của đồ thị sau:



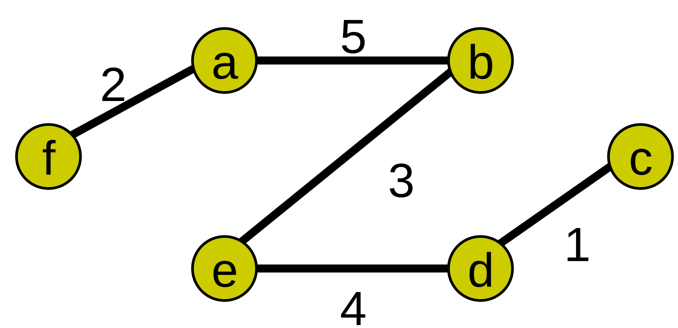
Dùng thuật toán Prim:

V_T	E_T	a	b	c	d	e	f
a	∅	0	5a	∞	∞	∞	2a
f	$\overline{af}$	-	5a	∞	∞	6f	-
b	$\overline{ab}$		-	4b	∞	3b	
e	$\overline{be}$			4b	4e	-	
d	$\overline{ed}$			1d	-		
c	$\overline{dc}$			-			

Vậy cây khung nhỏ nhất với tập cạnh

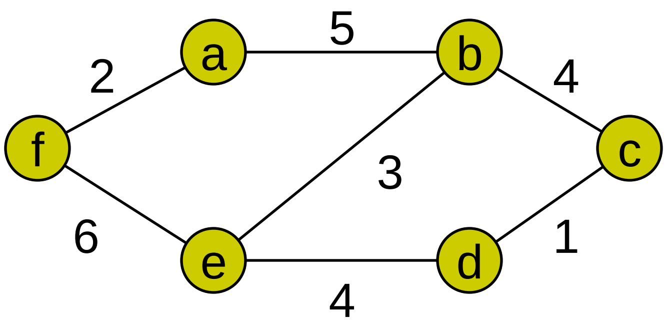
$$E_T = \overline{af} \cup \overline{ab} \cup \overline{be} \cup \overline{ed} \cup \overline{dc}$$

có độ dài (trọng số): 2+5+3+4+1=15



ây khung (Spanning Tree)

- Cây khung nhỏ nhất
  - **Ví dụ:** Tìm cây khung nhỏ nhất của đồ thị sau:



Dùng thuật toán Kruskal:

Sắp xếp các cạnh của đồ thị theo thứ tự có trọng số không giảm:

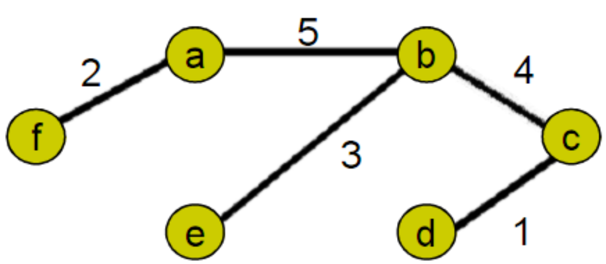
$\overline{cd}, \overline{af}, \overline{be}, \overline{bc}, \overline{ed}, \overline{ab}, \overline{ef}$

Bước chọn	E_T	Trọng số
1	$\overline{cd}$	1
2	$\overline{af}$	2
3	$\overline{be}$	3
4	$\overline{bc}$	4
5	$\overline{ab}$	5

Vậy cây khung nhỏ nhất với tập cạnh

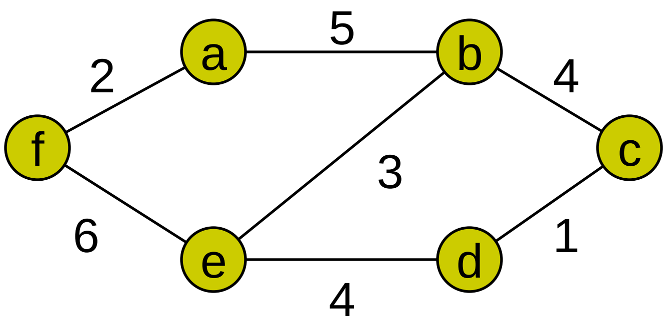
$$E_T = \overline{cd} \cup \overline{af} \cup \overline{be} \cup \overline{bc} \cup \overline{ab}$$

có độ dài (trọng số): 1+2+3+4+5 =15



ây khung (Spanning Tree)

- Cây khung nhỏ nhất
  - **Ví dụ:** Tìm cây khung nhỏ nhất của đồ thị sau:



Dùng thuật toán Kruskal:

Sắp xếp các cạnh của đồ thị theo thứ tự có trọng số không giảm:

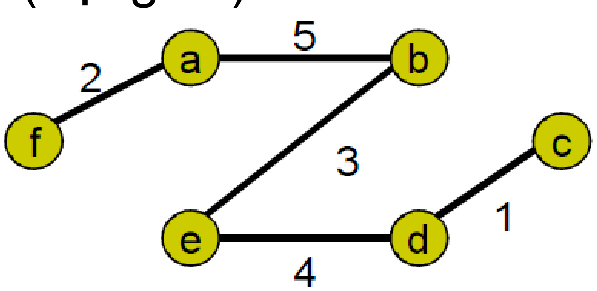
$$\overline{cd}\ ,\ \overline{af}\ ,\ \overline{be}\ ,\ \overline{bc}\ ,\ \overline{ed}\ ,\ \overline{ab},\ \overline{ef}$$

Bước chọn	E_T	Trọng số
1	$\overline{cd}$	1
2	$\overline{af}$	2
3	$\overline{be}$	3
4	$\overline{ed}$	4
5	$\overline{ab}$	5

Vậy cây khung nhỏ nhất với tập cạnh

$$E_T = \overline{cd} \cup \overline{af} \cup \overline{be} \cup \overline{ed} \cup \overline{ab}$$

có độ dài (trọng số): 1+2+3+4+5 =15



Cây khung (Spanning Tree)

- Cây khung nhỏ nhất
  - So sánh Prim và Kruskal
    - Prim chọn cạnh có trọng số nhỏ nhất liên thuộc với một đỉnh đã thuộc cây và không tạo ra chu trình
    - Kruskal chọn cạnh có trọng số nhỏ nhất miễn là không tạo ra chu trình
    - Thuật toán Prim hiệu quả hơn đối với các đồ thị dày (số cạnh nhiều)

Cây khung (Spanning Tree)

- Một số bài toán ứng dụng
  - **Nối dây điện**
    - Trong một mặt phẳng toạ độ cho N + 1 điểm, điểm đầu tiên chính là gốc tọa độ được coi là nguồn điện duy nhất mà từ đó ta nối dây cấp điện cho các nơi khác. Điểm thứ i trong N điểm còn lại có toạ độ là (Xi, Yi), là điểm đặt máy thứ i. Mỗi điểm đặt máy có thể lấy trực tiếp từ nơi cấp điện ban đầu hay gián tiếp qua một điểm đặt máy khác.
    - Yêu cầu đưa ra phương án nối điện giữa các điểm để mọi nơi đặt máy đều có điện và tổng chiều dài dây cần thiết là ngắn nhất.

Cây khung (Spanning Tree)

- Một số bài toán ứng dụng
  - Theo thiết kế, một mạng giao thông gồm N nút. Biết trước chi phí để xây dựng đường hai chiều trực tiếp từ nút *i* đến nút *j*. Hai tuyến đường khác nhau không cắt nhau tại điểm không là đầu mút. Hiện đã xây dựng được K tuyến đường.
  - *Bài toán :* Hệ thống đường đã xây dựng đã bảo đảm sự đi lại giữa hai nút bất kỳ chưa? Nếu chưa, hãy chọn một số tuyến đường cần xây dựng thêm sao cho:
    - Các tuyến đường sẽ xây dựng thêm cùng với các đường đã xây dựng bảo đảm sự đi lại giữa hai nút bất kỳ.
    - Tổng kinh phí xây dựng các tuyến đường thêm vào là ít nhất.

Cây khung (Spanning Tree)

- **Cây khung lớn nhất**
  - **Định nghĩa**
    - Cây khung lớn nhất trong một đồ thị liên thông, có trọng số là một cây khung có tổng trọng số trên các cạnh của nó là lớn nhất.

Tương tự trình bày thuật toán Prim và Kruskal để tìm cây khung lớn nhất trong đồ thị liên thông có trọng số !!!