Chương 6: Lý thuyết đồ thị

Giảng viên: Phạm Thị Lan
Bộ môn: Khoa học máy tính

CÂY VÀ ỨNG DỤNG

NỘI DUNG

1. Cây ☞
2. Cây nhị phân tìm kiếm ☞
3. Cây biểu thức số học ☞
4. Duyệt cây nhị phân ☞

1. CÂY

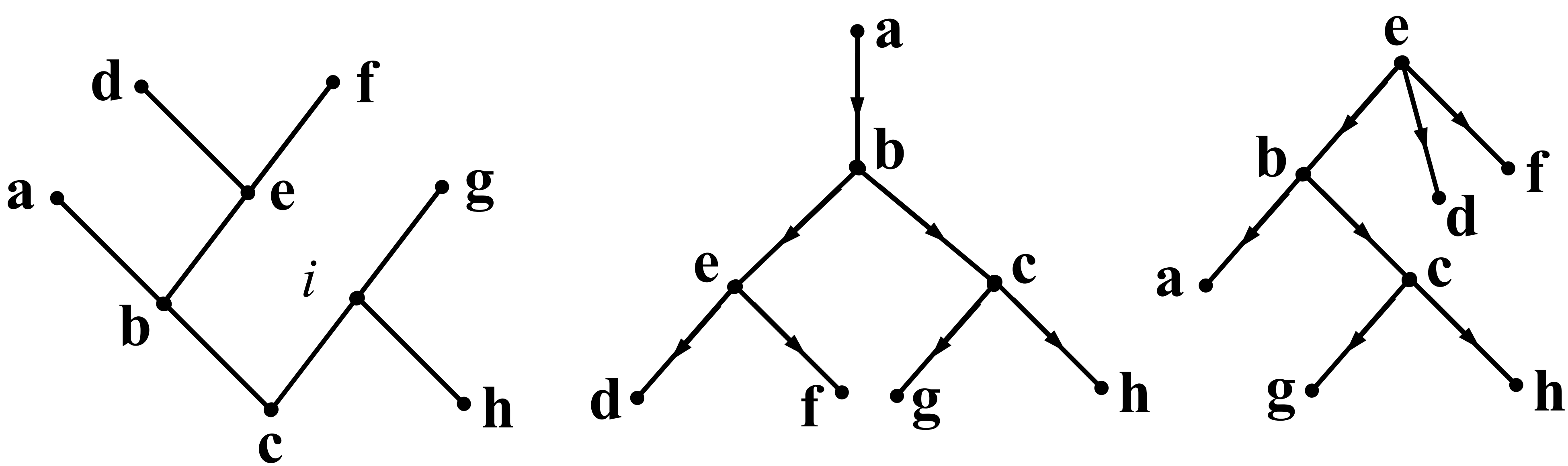
Định nghĩa: Cây là một đơn đồ thị vô hướng, liên thông và không có chu trình.

Cây có gốc: nếu ta chọn một đỉnh đặc biệt gọi là gốc của cây và định hướng các cạnh trên cây từ gốc đi ra thì ta được một đồ thị có hướng gọi là cây có gốc.

Chọn đỉnh làm gốc khác nhau sẽ tạo ra các cây khác nhau.

1. CÂY

Ví dụ:



1. CÂY

Một số khái niệm: Cho T là một cây có gốc, v là một đỉnh khác gốc của T.

+ *Cha* của v là đỉnh u nếu có một cạnh có hướng duy nhất từ u → v. Khi đó, u được gọi là cha của v; v là con của u.

+ *Anh em* là các đỉnh có cùng cha.

+ *Tổ tiên* của một đỉnh khác gốc là các đỉnh trên đường đi từ gốc đến đỉnh đó.

+ *Con cháu* của v là các đỉnh có v là tổ tiên.

+ *Lá* là đỉnh không có con.

+ *Đỉnh trong* là các đỉnh có con.

1. CÂY

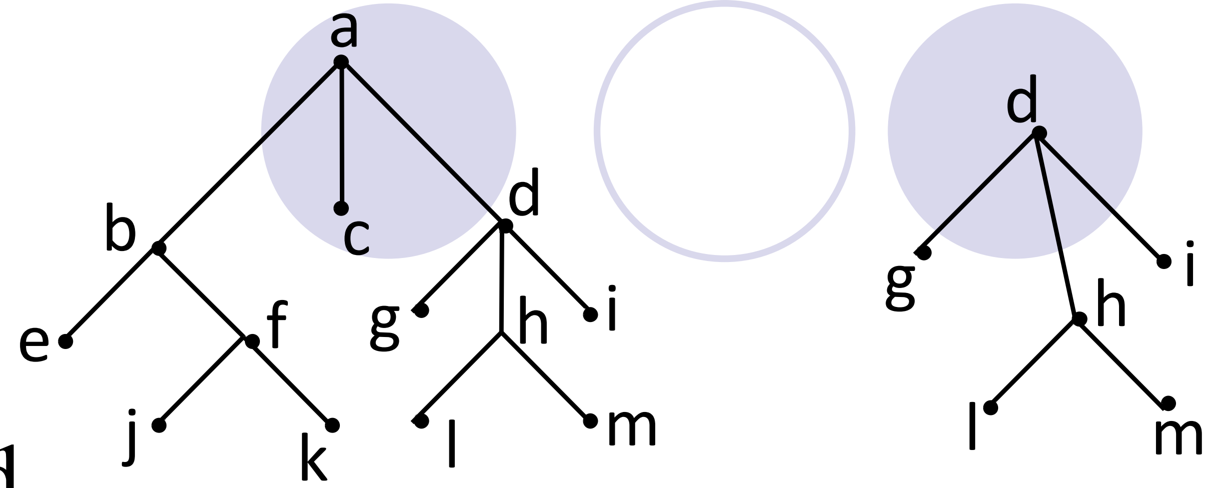
+ *Cây con* với gốc a là đồ thị con của cây đang xét, bao gồm a và các con cháu của nó cùng tất cả các cạnh liên thuộc với các con cháu của a.

+ *Mức của một đỉnh*: là khoảng cách từ gốc đến nó.

+ *Chiều cao của cây*: mức lớn nhất của một đỉnh bất kỳ trong cây gọi là chiều cao của cây.

1. CÂY

Ví dụ:



- *Con* của a là b, c, d
- *Cha* của b, c, d là a
- b, c, d là *anh em*.
- *Con cháu* của d là g, h, i, l, m
- *Tổ tiên* của g là a, d.
- a, b, d, f, h là *đỉnh trong*

- e, c, j, k, l, m, i là *lá*
- Cây bên phải là *cây con* gốc d của cây bên trái
- *Mức* của b, c, d là 1. Mức của e, f, g, h, i là 2.
- *Chiều cao* của cây là 3

1. CÂY

Định lí 1: Một cây bất kì với ít nhất 2 đỉnh thì có ít nhất hai đỉnh treo.

Định lí 2: Một cây có n đỉnh có đúng n – 1 cạnh.

1. CÂY

Định lý Daisy Chain: Cho T là một đồ thị có n đỉnh. Khi đó các khẳng định sau là tương đương:

(1): T là một cây.

(2): T không có chu trình và có n - 1 cạnh.

(3): T là một đồ thị liên thông và nếu hủy bất kỳ một cạnh nào của nó cũng làm mất tính liên thông.

(4): Giữa 2 đỉnh bất kỳ của T, luôn tồn tại một đường đi đơn duy nhất nối 2 đỉnh này.

(5): T không có chu trình và nếu thêm một cạnh mới nối 2 đỉnh bất kỳ của T thì sẽ tạo ra một chu trình.

(6): T liên thông và có n - 1 cạnh.

2. CÂY NHỊ PHÂN TÌM KIẾM

Định nghĩa cây nhị phân: là một cây có gốc. Mỗi đỉnh trong của cây không có quá hai con, trong đó có một con bên trái và con bên phải.

Cây nhị phân được gọi là đầy đủ nếu mỗi đỉnh trong của nó có đúng 2 con.

Cây nhị phân tìm kiếm: là một cây nhị phân thoả mãn:

(1) Mỗi đỉnh được gán một khoá khác nhau.

(2) Khoá của một đỉnh trong lớn hơn khoá con trái và nhỏ hơn khoa con phải của nó.

2. CÂY NHỊ PHÂN TÌM KIẾM

Thuật toán xây dựng cây nhị phân tìm kiếm cho một dãy khoá.

(1) Chọn khoá v_0 làm gốc.

(2) [Lặp] với mỗi khoá v còn lại thực hiện như sau:

+ $v_0 \leftarrow$ gốc.

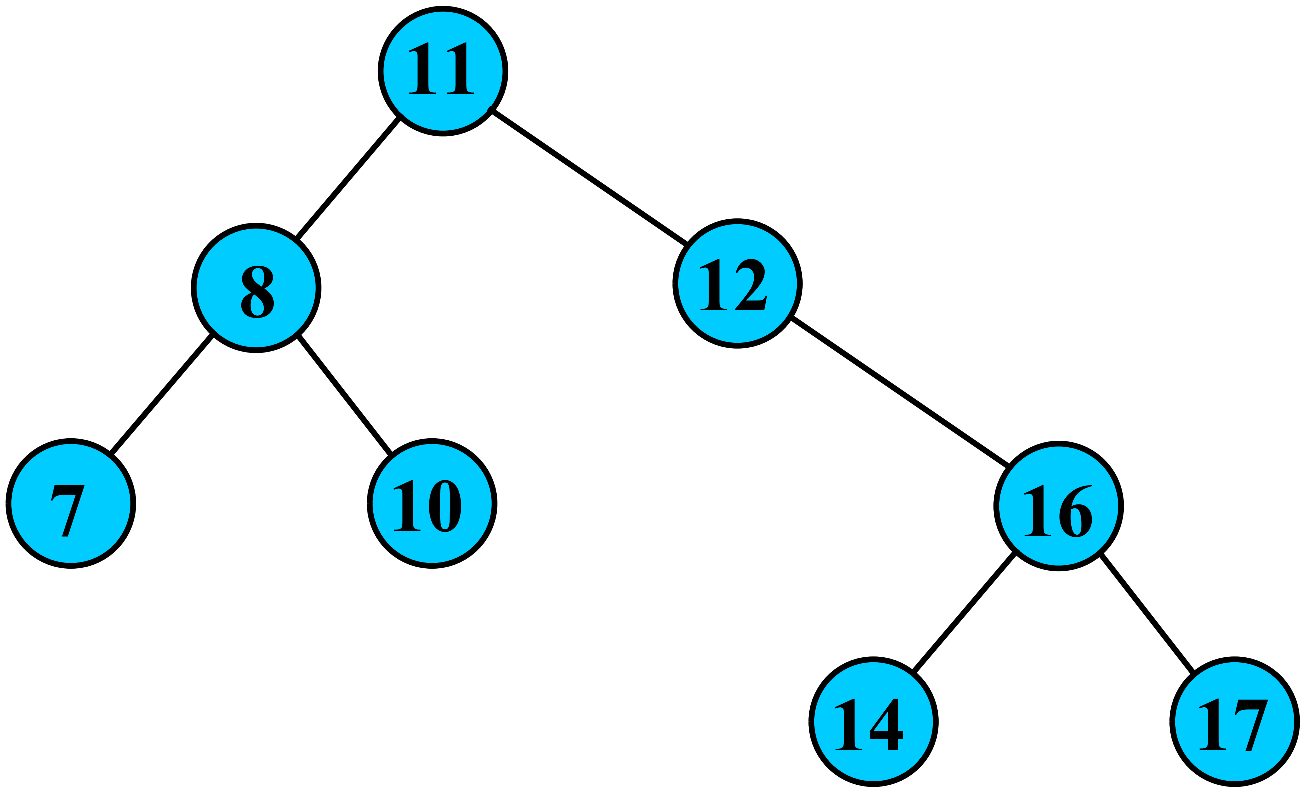
+ $v > v_0$: Nếu v_0 đã có con phải thì $v_0 \leftarrow$ con phải
nếu không thì v là con phải của v_0.

+ $v < v_0$: Nếu v đã có con trái thì $v_0 \leftarrow$ con trái
nếu không thì v là con trái của v_0.

Thuật toán kết thúc khi mọi khoá đã được đưa vào cây.

2. CÂY NHỊ PHÂN TÌM KIẾM

__Ví dụ:__ 11, 8, 7, 10, 12, 16, 14, 17



2. CÂY NHỊ PHÂN TÌM KIẾM

Ví dụ: 11, 8, 7, 10, 12, 16, 14, 17

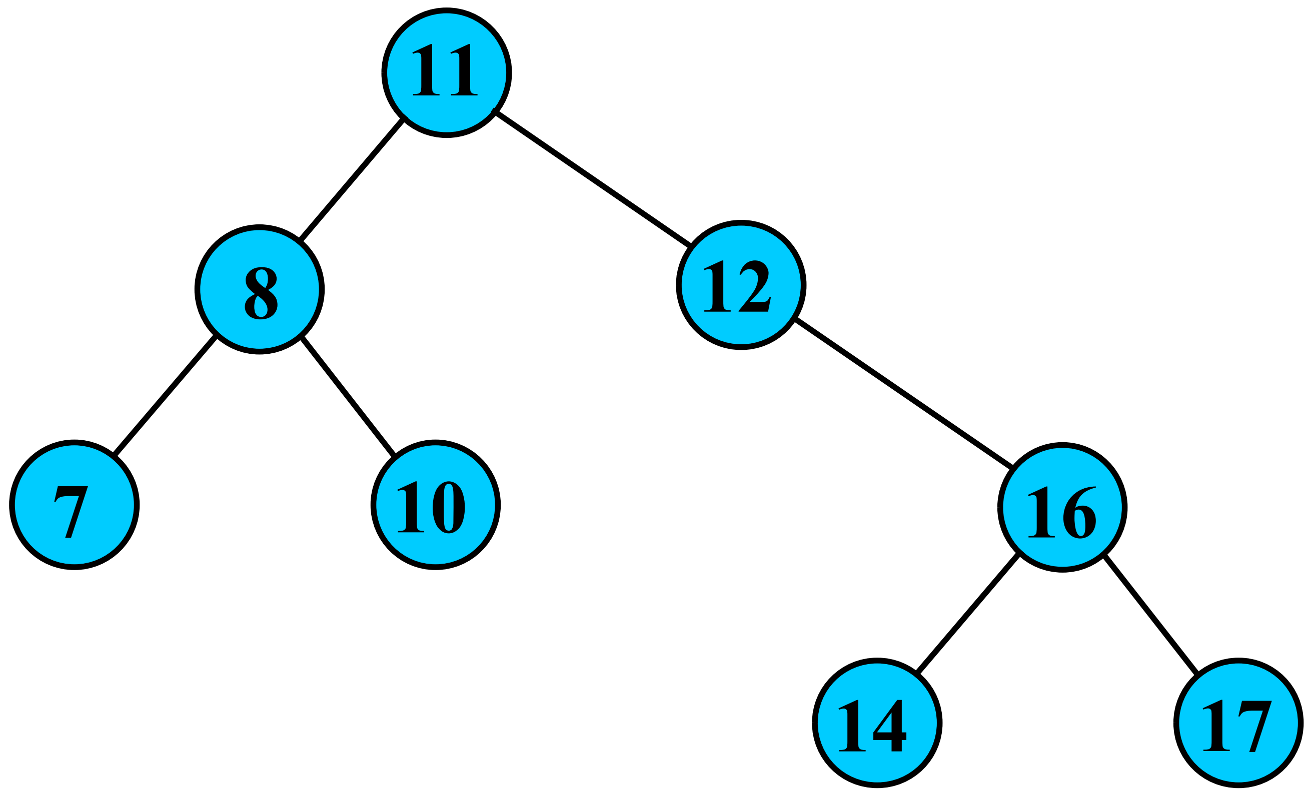
Tìm khoá 16:

+ So sánh 11.

+ so sánh 12.

+ So sánh 16.

16 có trong dãy đã cho



2. CÂY NHỊ PHÂN TÌM KIẾM

<u>Ví dụ:</u> 11, 8, 7, 10, 12, 16, 14, 17

Tìm khoá 15:

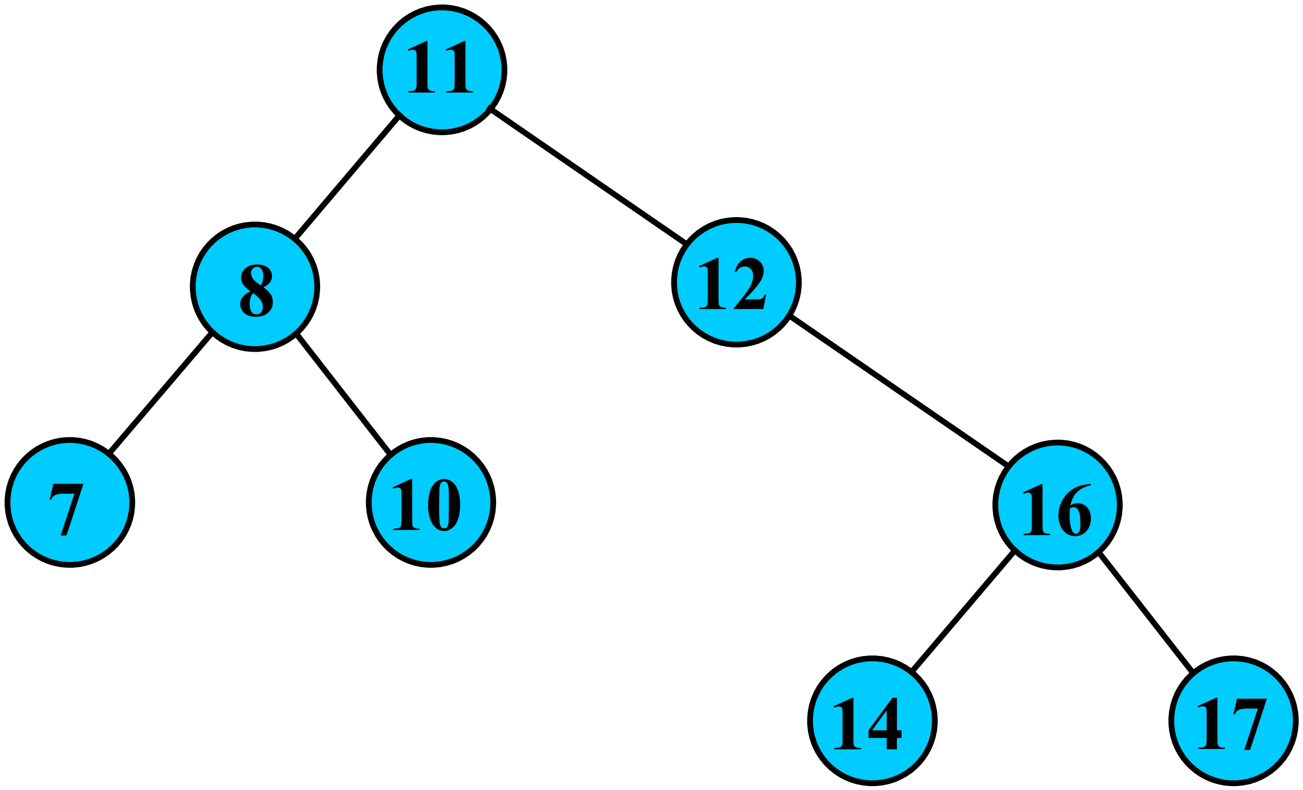
+ So sánh 11.

+ so sánh 12.

+ So sánh 16.

+ So sánh 14.

15 không có trong dãy đã cho



3. CÂY BIỂU THỨC SỐ HỌC

Cây biểu diễn biểu thức số học: là một cây nhị phân thoả mãn:

(1) Nút trong biểu diễn toán tử 2 ngôi θ (+ - * / ^).

(2) Lá biểu diễn cho một toán hạng của biểu thức.

Biểu thức E_1 θ E_2 được biểu diễn bởi cây có:

+ Gốc biểu diễn θ.

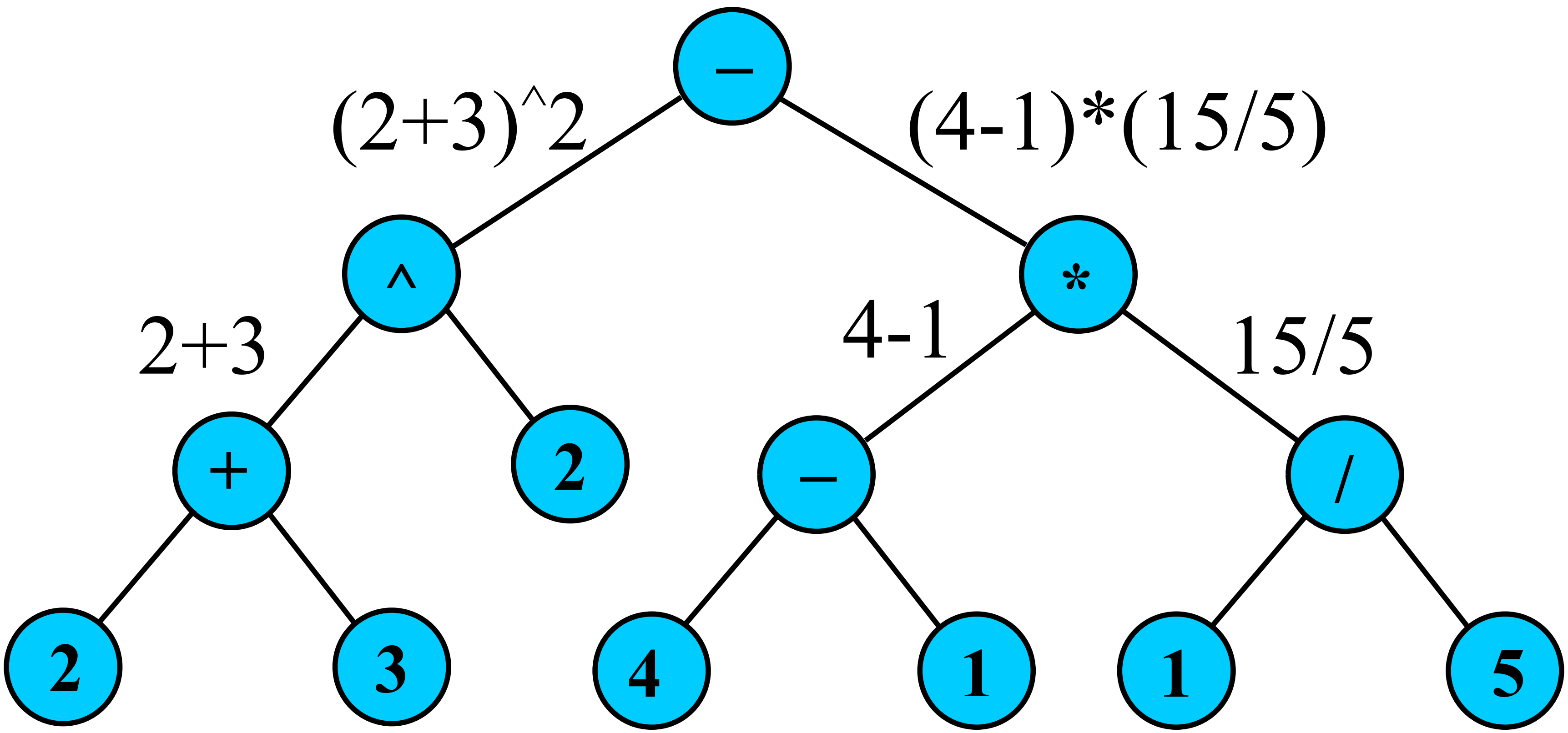
+ Con trái biểu diễn cho biểu thức E_1.

+ Con phải biểu diễn cho biểu thức E_2.

3. CÂY BIỂU THỨC SỐ HỌC

<u>*Ví dụ:*</u>

$$E = (2+3)\hat{}2 - (4-1)*(15/5)$$



4. DUYỆT CÂY NHỊ PHÂN

3 phương pháp duyệt cây nhị phân

\+ Duyệt tiền tự (PreOrder):

(1): Duyệt nút gốc.

(2): Duyệt con trái theo phương pháp tiền tự.

(3): Duyệt con phải theo phương pháp tiền tự.

\+ Duyệt trung tự (InOrder):

(1): Duyệt con trái theo phương pháp trung tự.

(2): Duyệt nút gốc.

(3): Duyệt con phải theo phương pháp trung tự.

4. DUYỆT CÂY NHỊ PHÂN

+ *Duyệt hậu tự (PostOrder)*:

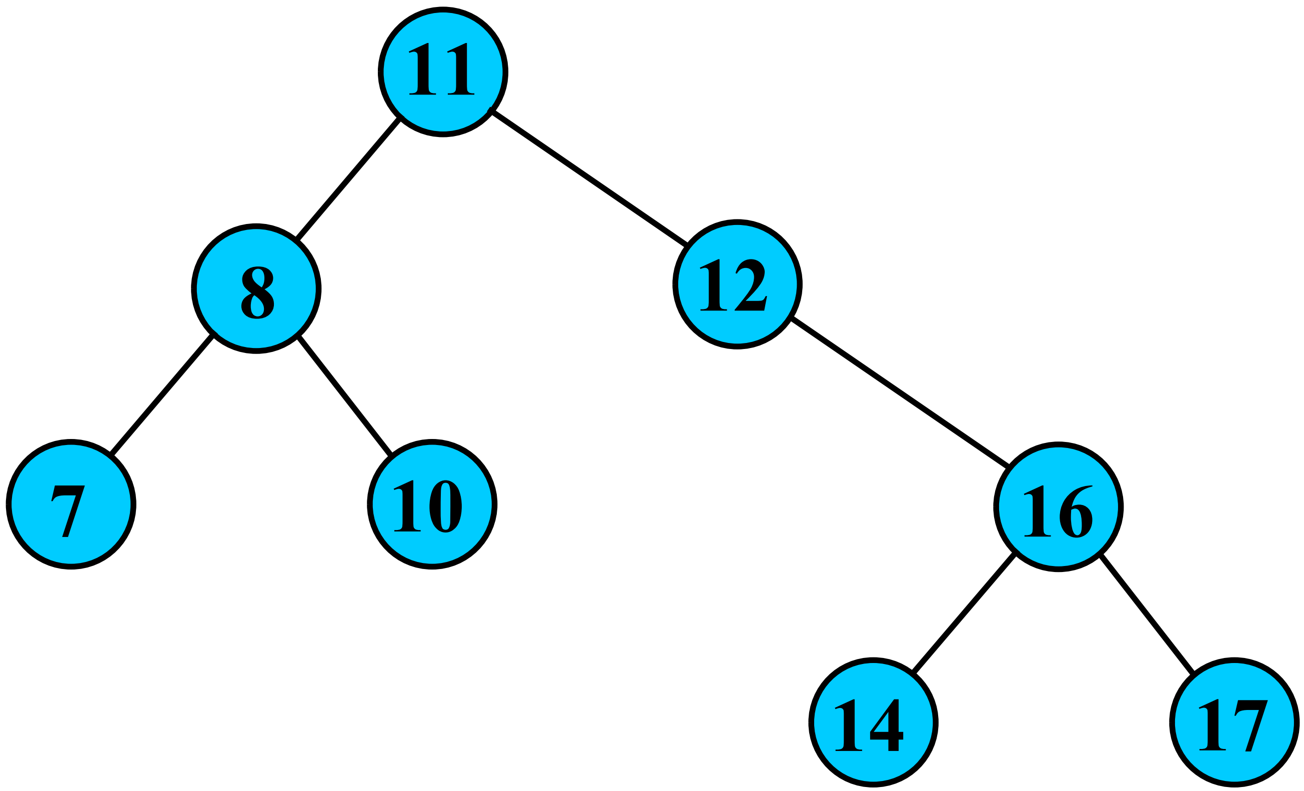
(1): Duyệt con trái theo phương pháp hậu tự.

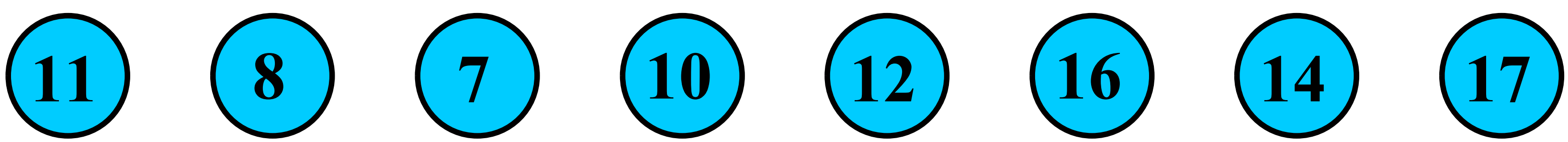
(2): Duyệt con phải theo phương pháp hậu tự.

(3): Duyệt nút gốc.

4. DUYỆT CÂY NHỊ PHÂN

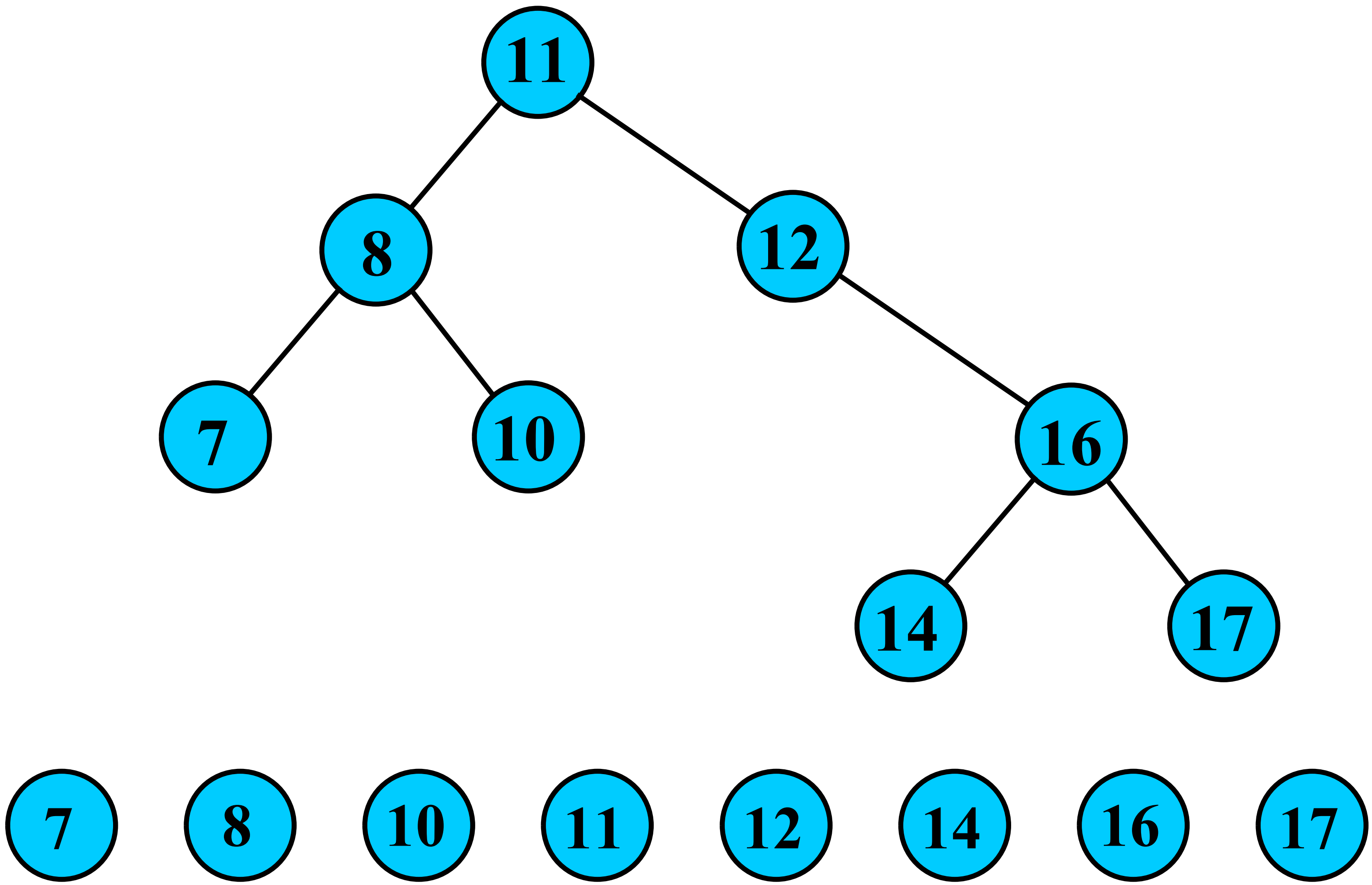
Ví dụ: Duyệt cây nhị phân sau theo phương pháp tiền tự





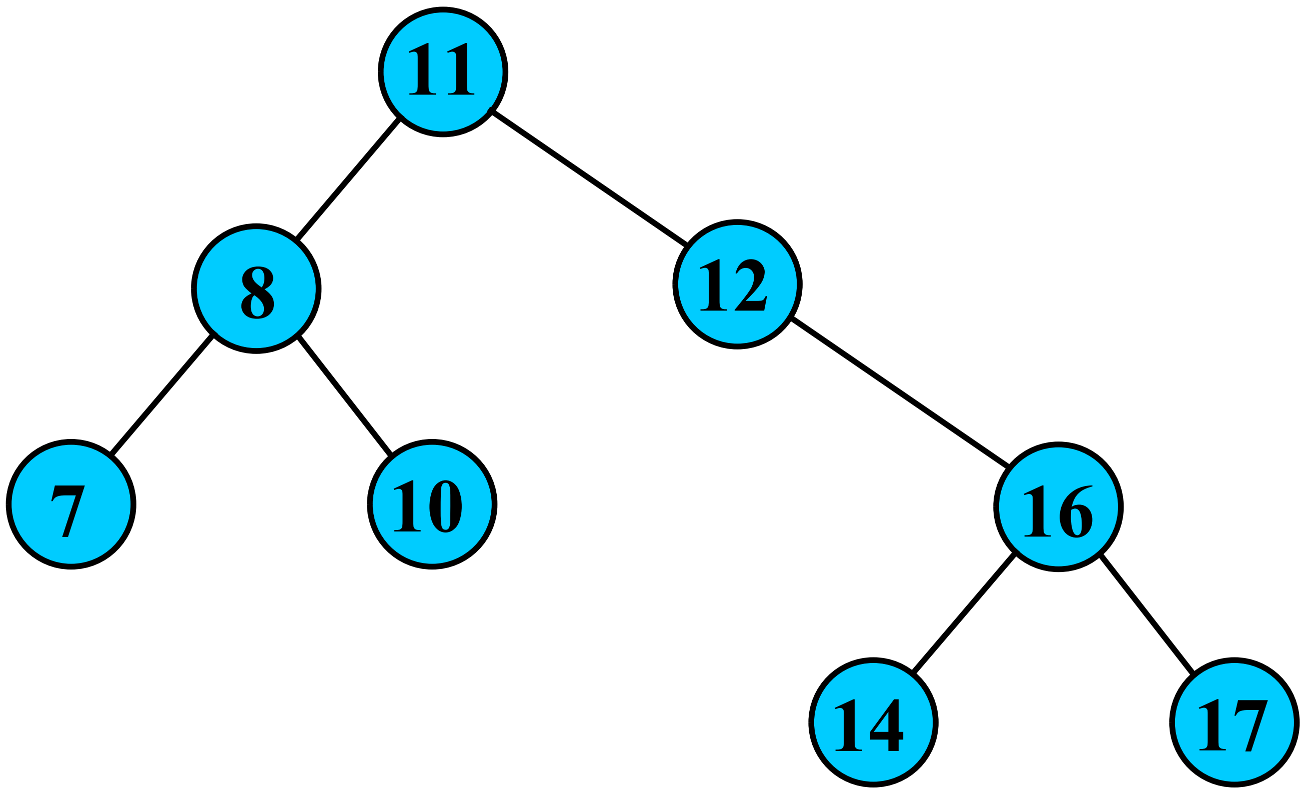
4. DUYỆT CÂY NHỊ PHÂN

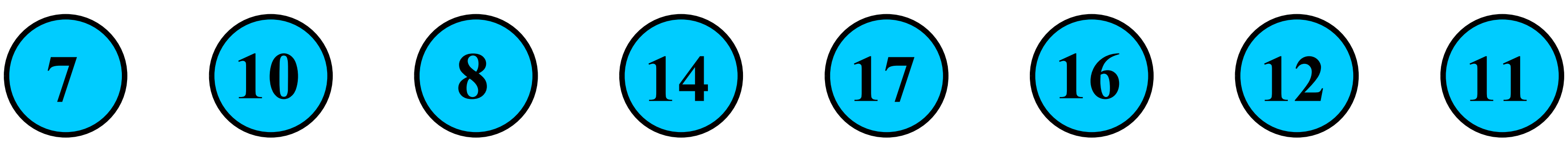
Ví dụ: Duyệt cây nhị phân sau theo phương pháp trung tự



4. DUYỆT CÂY NHỊ PHÂN

Ví dụ: Duyệt cây nhị phân sau theo phương pháp hậu tự





CÂY KHUNG CỦA ĐỒ THỊ

NỘI DUNG

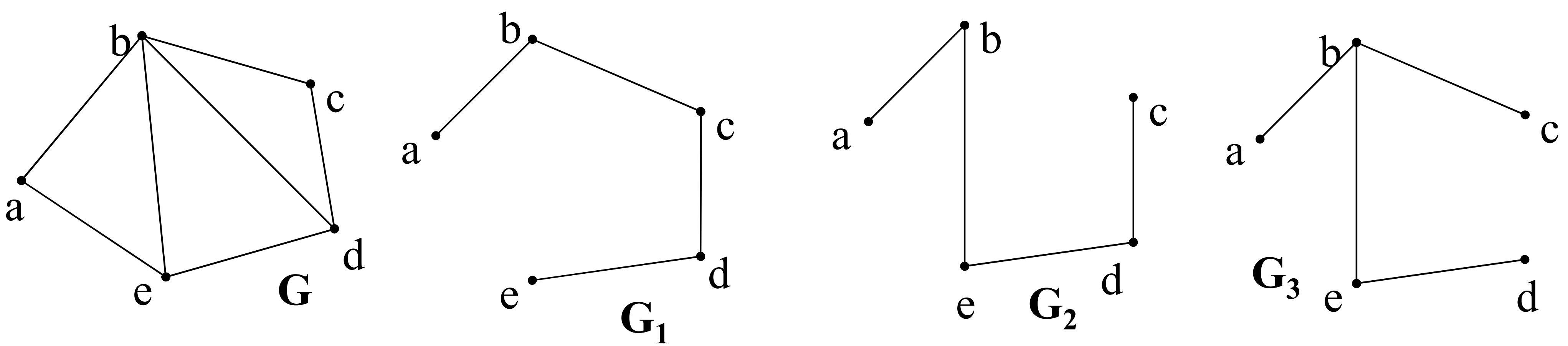
1. Định nghĩa ☞
2. Xác định cây khung ☞
   2.1. Theo chiều rộng
   2.2. Theo chiều sâu
3. Cây khung nhỏ nhất ☞

1. ĐỊNH NGHĨA

Định nghĩa cây khung: Cho đồ thị G = (V, E) là một đồ thị vô hướng liên thông.

Cây T = (V, F) với F là tập con của E được gọi là cây khung của đồ thị G.

Ví dụ:



2. XÁC ĐỊNH CÂY KHUNG

Xác định cây khung là việc xây dựng một cây chứa tất cả các đỉnh của đồ thị.

Hai thuật toán xác định cây khung là:

+ Xác định ưu tiên theo chiều rộng.

+ Xác định ưu tiên theo chiều sâu.

2. XÁC ĐỊNH CÂY KHUNG

2.1. Theo chiều sâu.

+ *Bước 1*: Lấy một đỉnh a làm gốc của cây khung.

+ *Bước 2*: Xây dựng đường đi từ đỉnh này bằng cách ghép lần lượt các cạnh vào. Mỗi cạnh được ghép vào nối đỉnh cuối cùng của đường đi và một đỉnh chưa có trong cây. Thực hiện đến khi không ghép được thêm cạnh nào nữa.

+ *Bước 3*: Nếu đường đi chứa tất cả các đỉnh của đồ thị thì đó chính là cây khung. Nếu không thì chuyển sang bước 4.

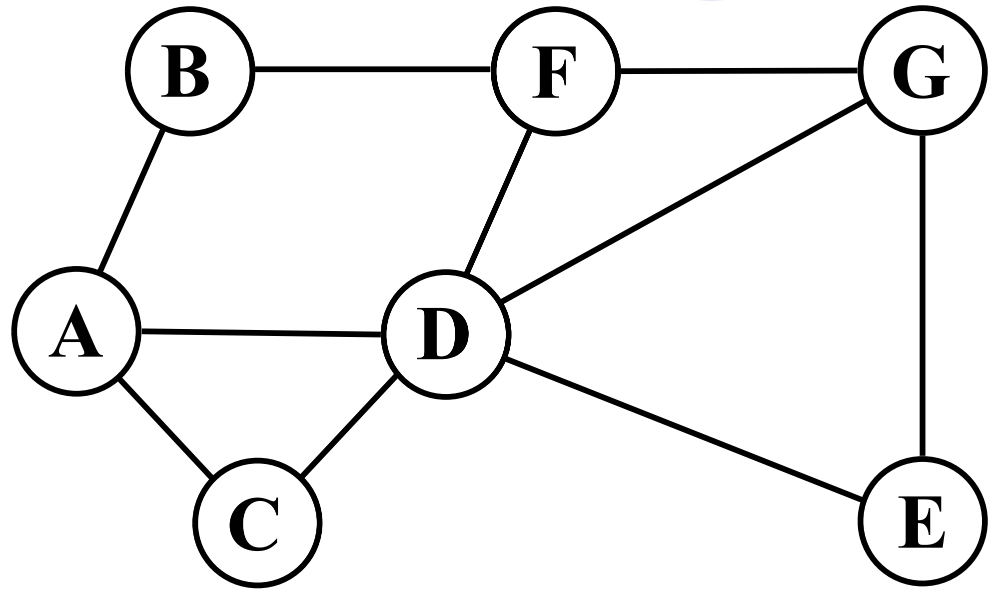
+ *Bước 4*: Quay lui lại đỉnh ngay trước đỉnh cuối cùng của đường đi và xây dựng đường đi mới bắt đầu từ đỉnh này. Nếu không được thì lùi tiếp đỉnh nữa.

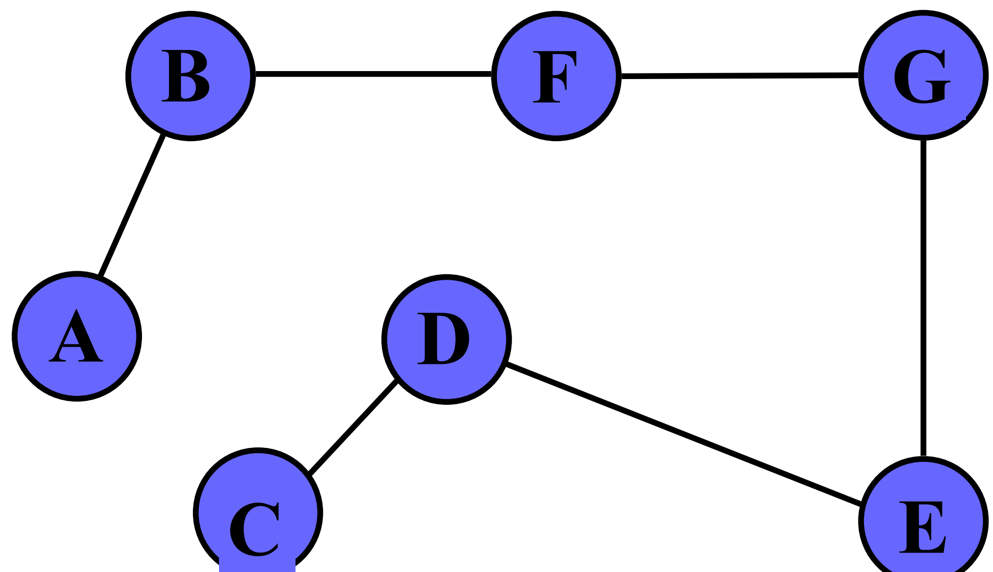
2. XÁC ĐỊNH CÂY KHUNG

2.1. Theo chiều sâu

<u>Ví dụ:</u>

TT	Cạnh ghép vào





2. XÁC ĐỊNH CÂY KHUNG

2.2. Theo chiều rộng

Các bước thực hiện của BFS(v)

+ *Bước 1*: Chọn đỉnh a làm gốc của cây.

+ *Bước 2*: ghép các cạnh liên thuộc với gốc. Các đỉnh kề với gốc trong bước này có mức là 1.

+ *Bước 3*: tiếp tục ghép các cạnh liên thuộc đỉnh mức 1 sao cho không tạo chu trình. Các đỉnh được đưa vào ở bước này có mức là 2.

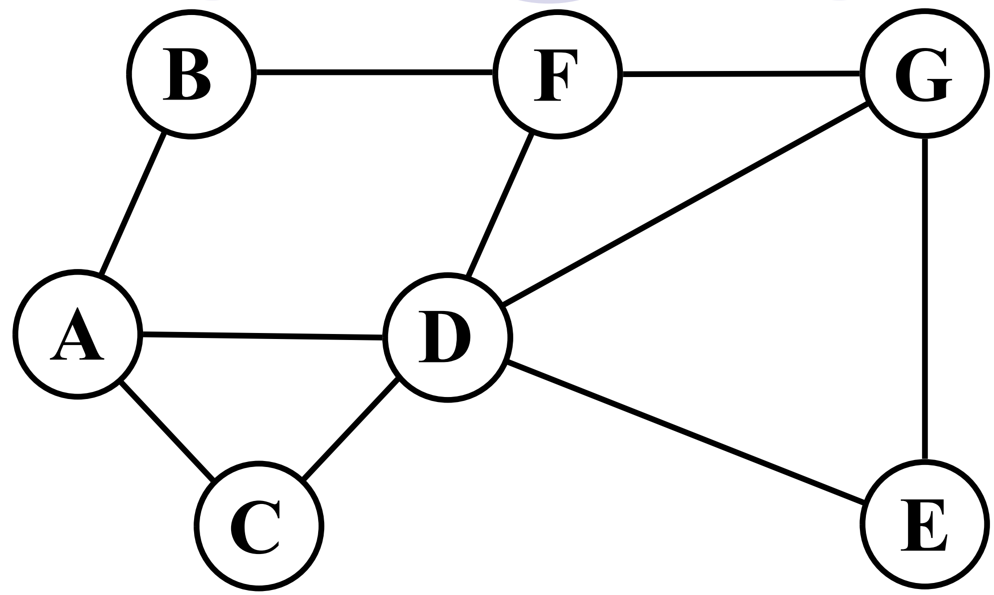
+ *Bước 4*: Tiếp tục quá trình khi tất cả các đỉnh đã được ghép vào cây.

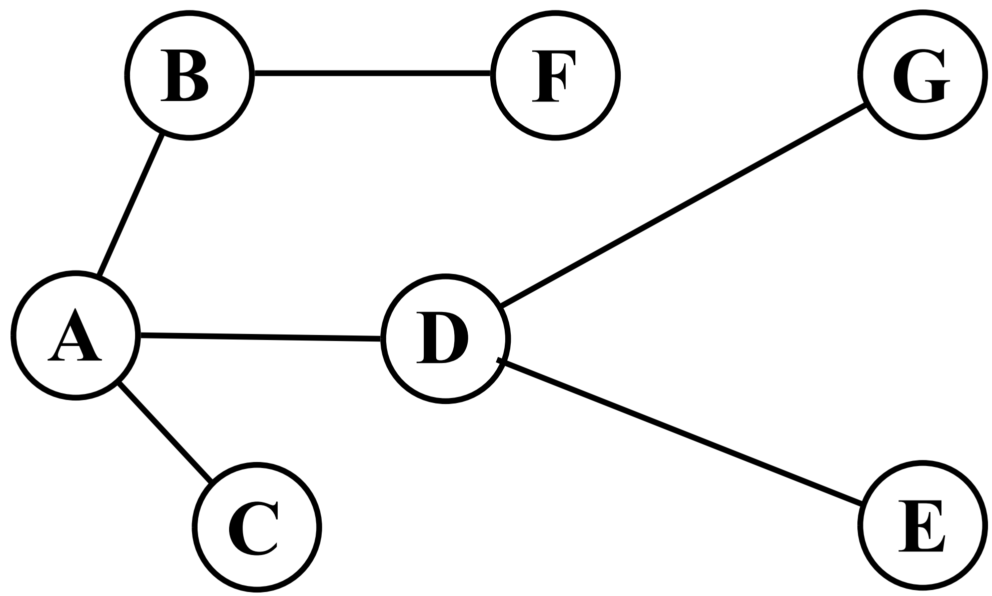
2. XÁC ĐỊNH CÂY KHUNG

2.1. Theo chiều sâu

Ví dụ:

TT	Cạnh ghép vào





2. XÁC ĐỊNH CÂY KHUNG

Ứng dụng hai thuật toán trên:

- Tìm đường đi giữa hai đỉnh. Đường đi xây dựng khi ưu tiên theo chiều rộng là đường đi ngắn nhất.

- Tìm các thành phần liên thông của đồ thị.

- Thuật toán theo chiều sâu để tìm chu trình.

3. CÂY KHUNG NHỎ NHẤT

Định nghĩa cây khung nhỏ nhất:

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.

Thuật toán tìm cây khung nhỏ nhất:

+ Prim (Robert Prim - 1957)

+ Kruskal (Joseph Kruskal – 1965)

3. CÂY KHUNG NHỎ NHẤT

Thuật toán Prim:

Đồ thị G = (V, E) liên thông, có n đỉnh.

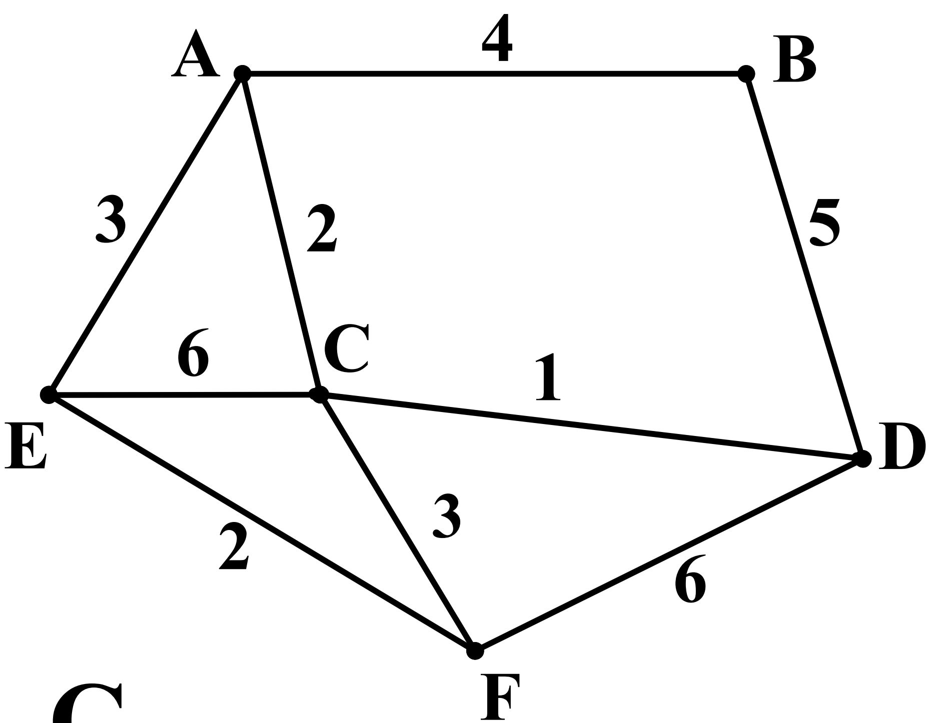
(1): Chọn một cạnh bất kỳ có trọng số nhỏ nhất, đặt nó vào cây khung.

(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.

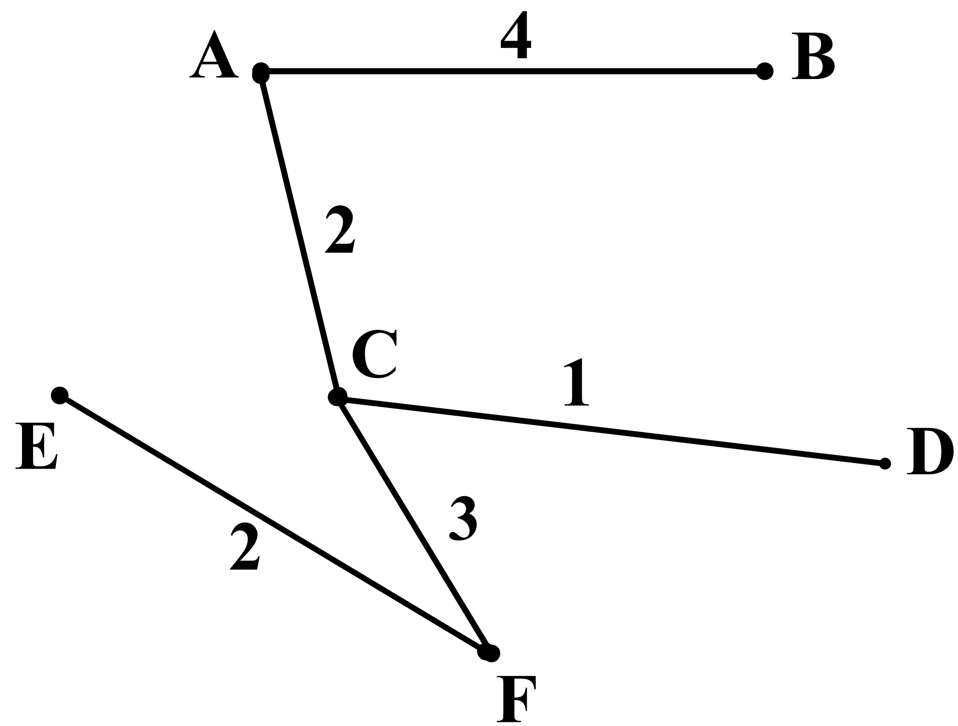
Thuật toán dừng lại khi n – 1 cạnh được ghép vào cây.

3. CÂY KHUNG NHỎ NHẤT

Ví dụ: Dùng thuật toán Prim tìm cây khung nhỏ nhất của đồ thị sau:



G



Cạnh đã được chọn

Cạnh có thể chọn

Cây khung của G

Trọng số: 12

3. CÂY KHUNG NHỎ NHẤT

Thuật toán Kruskal:

Đồ thị G = (V, E) liên thông, có n đỉnh.

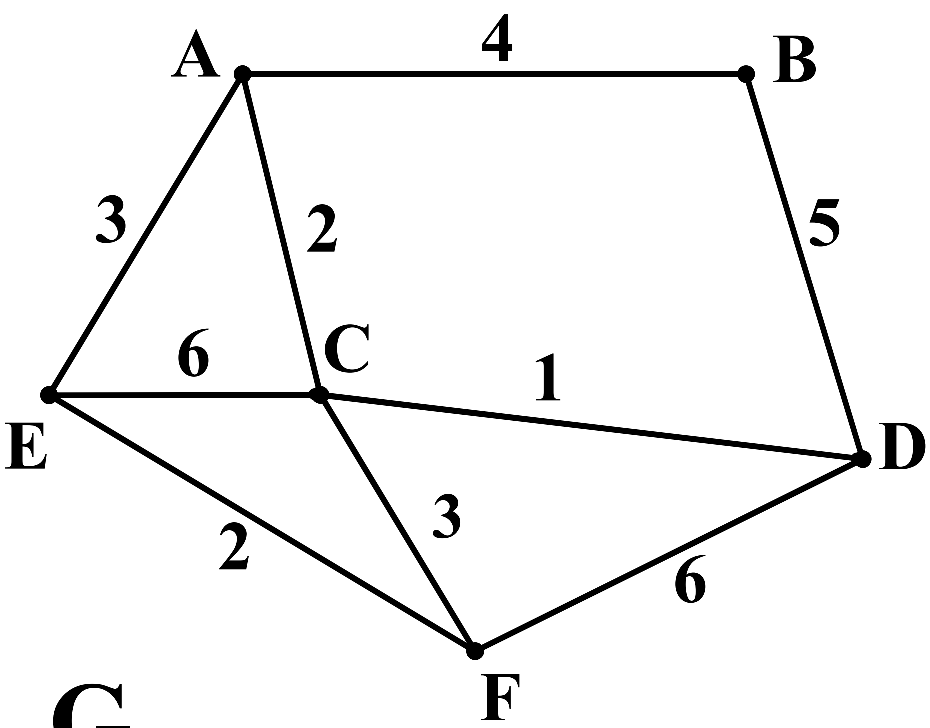
(1): Chọn một cạnh bất kỳ có trọng số nhỏ nhất, đặt nó vào cây khung.

(2): Lần lượt ghép vào cây các cạnh có trọng số nhỏ nhất mà không tạo ra chu trình trong cây.

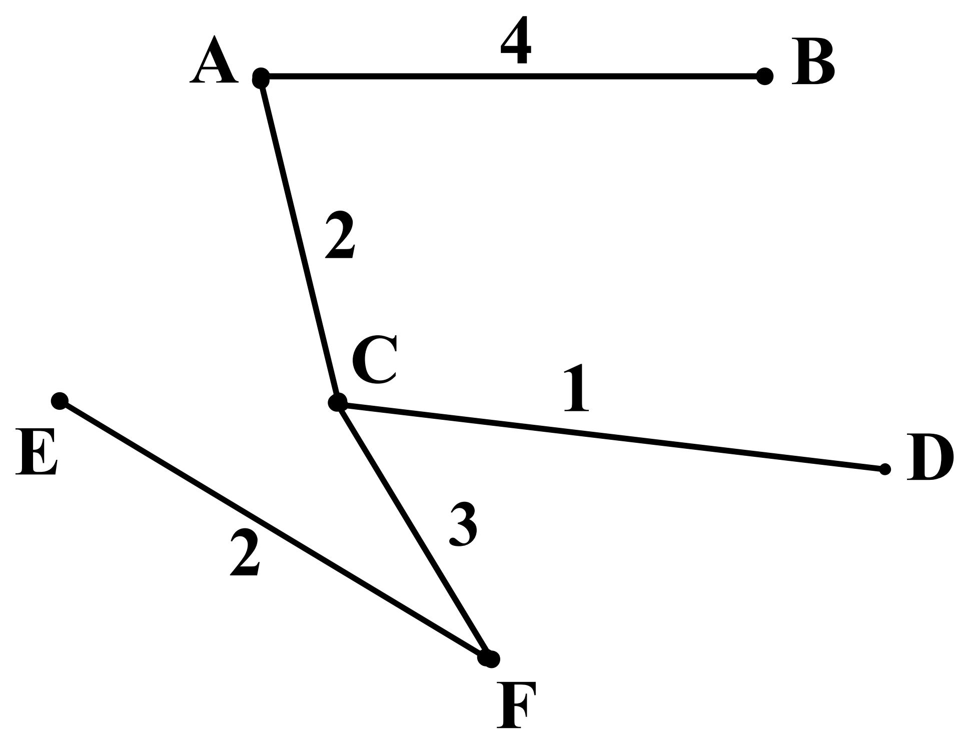
Thuật toán dừng lại khi n – 1 cạnh được ghép vào cây.

3. CÂY KHUNG NHỎ NHẤT

<u>*Ví dụ*</u>: Dùng thuật toán Kruskal tìm cây khung nhỏ nhất của đồ thị sau:



G



Cạnh đã được chọn

Cây khung của G

Trọng số: 12

ĐỒ THỊ CÓ HƯỚNG

NỘI DUNG

1. Bậc của đỉnh ☞
2. Biểu diễn đồ thị trên máy tính ☞
3. Dây chuyền, băng chuyền và chu trình ☞
4. Đồ thị có hướng liên thông ☞
5. Đồ thị phản chu trình ☞

1. BẬC CỦA ĐỈNH

Bậc của đỉnh: Cho G = [V, E], v $\in$V. Ta có:

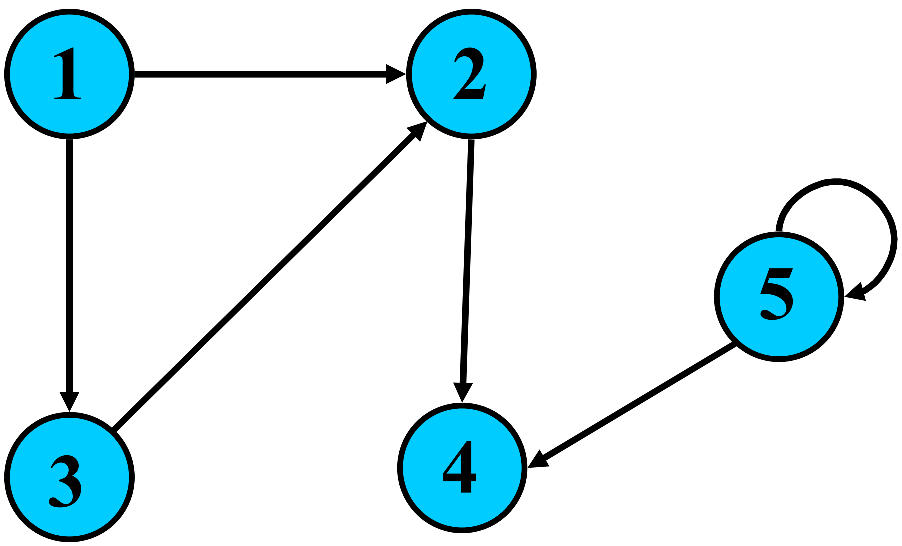
+ deg-(v) (bậc vào) là số cạnh nhận v là đỉnh cuối (đỉnh vào)

+ deg+(v) (bậc ra) là số cạnh nhận v là đỉnh đầu (đỉnh ra).

Một khuyên trên đồ thị có hướng sẽ đóng góp 2 đơn vị vào bậc vào và bậc ra tương ứng của đỉnh đó.

1. BẬC CỦA ĐỈNH

Ví dụ:



$deg^{+}(1) = 2,\ deg^{-}(1) = 0$

$deg^{+}(2) = 1,\ deg^{-}(2) = 2$

$deg^{+}(3) = 1,\ deg^{-}(3) = 1$

$deg^{+}(4) = 0,\ deg^{-}(4) = 2$

$deg^{+}(5) = 2,\ deg^{-}(5) = 1$

Chú ý:

$deg^{+}(v) = 0 \rightarrow$ v là đỉnh hạ lưu.

$deg^{-}(v) = 0 \rightarrow$ v là đỉnh nguồn.

1. BẬC CỦA ĐỈNH

<u>*Định lý*</u>: Giả sử G = [V, E] là một đồ thị có hướng có m cung khi đó:

$$m = \sum_{v \in V} \deg^{-}(v) = \sum_{v \in V} \deg^{+}(v)$$

2. BIỂU DIỄN ĐỒ THỊ TRÊN MÁY TÍNH

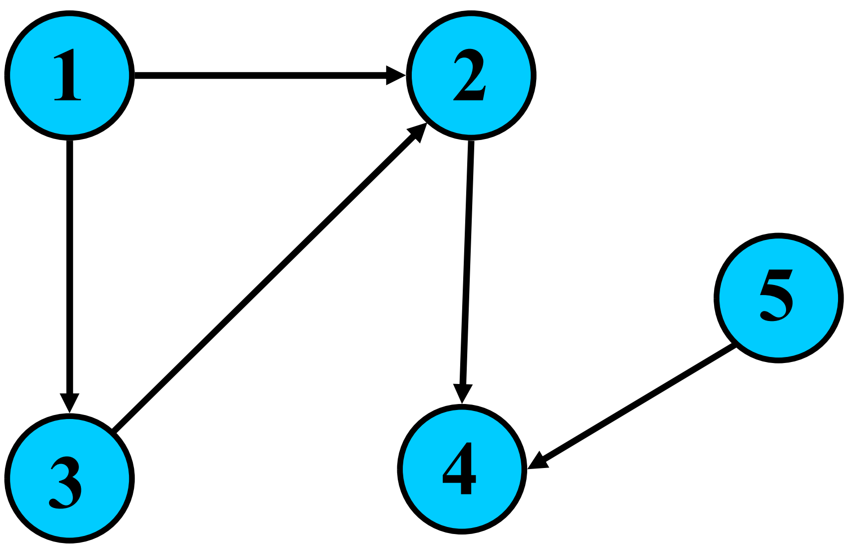
Ma trận kề: Cho đơn đồ thị $G = [V, E]$

với $V = \{v_1, \ldots, v_n\}$

Ma trận kề $A = [a_{ij}]_{n \times n}$ biểu diễn G.

a_{ij} là số cung đi từ v_i đến v_j

Ví dụ:



$$\begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} \begin{array}{c} \begin{array}{ccccc} \mathbf{1} & \mathbf{2} & \mathbf{3} & \mathbf{4} & \mathbf{5} \end{array} \\ \left(\begin{array}{ccccc} \quad & \quad & \quad & \quad & \quad \\ & & & & \\ & & & & \\ & & & & \\ & & & & \end{array} \right) \end{array}$$

2. BIỂU DIỄN ĐỒ THỊ TRÊN MÁY TÍNH

***Ma trận kề liên thuộc*:**

Cho đơn đồ thị $G = [V,E]$

với $V = \{v_1, \ldots, v_n\}$

$E = \{e_1, e_2, \ldots, e_m\}$

Ma trận kề $M = [m_{ij}]_{n \times m}$ biểu diễn G được xác định như sau:

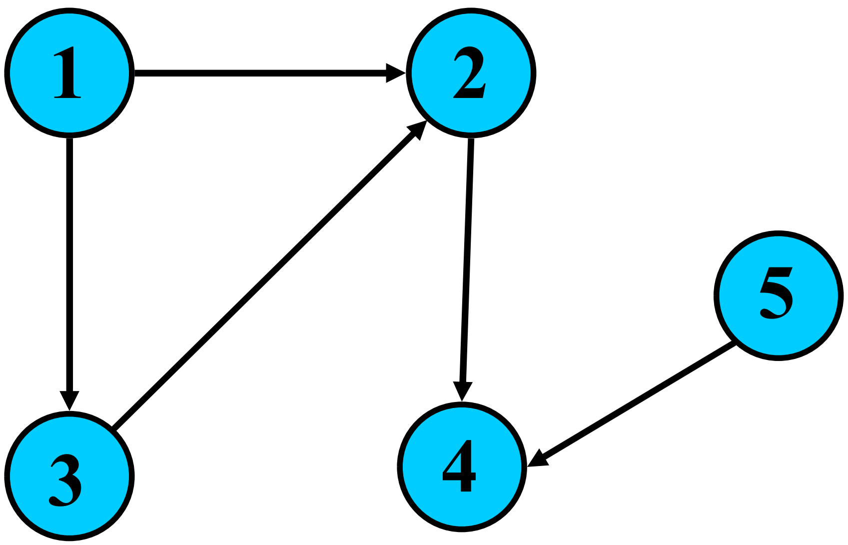
+ $m_{ij} = 1$ nếu v_i là đỉnh ra e_j

+ $m_{ij} = -1$ nếu v_i là đỉnh vào e_j

+ $m_{ij} = 0$ nếu e_j là không liên thuộc v_i

2. BIỂU DIỄN ĐỒ THỊ TRÊN MÁY TÍNH

Ví dụ:



$$
\begin{array}{c|ccccc}
 & [1,2] & [1,3] & [2,4] & [3,2] & [5,4] \\
\hline
1 & & & & & \\
2 & & & & & \\
3 & & & & & \\
4 & & & & & \\
5 & & & & &
\end{array}
$$

3. DÂY CHUYỀN, BĂNG CHUYỀN VÀ CHU TRÌNH

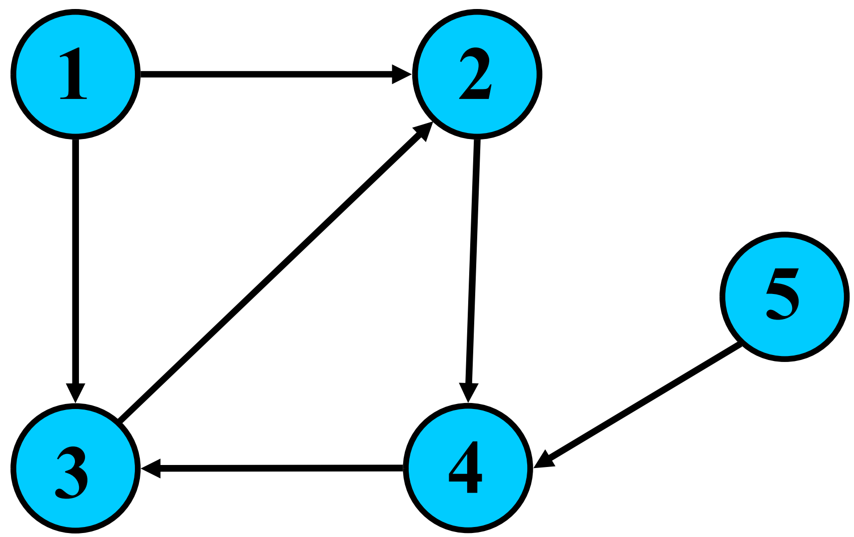
Dây chuyền: là một dãy các cung e_1, e_2, …, e_k sao cho khi bỏ hướng đi thì nó trở thành một dãy cạnh kế tiếp trong đồ thị vô hướng thu được

Băng chuyền: là một dây chuyền e_1, e_2, …, e_k mà đỉnh đích của cung e_i là đỉnh xuất phát của cung e_{i+1} với i = 1,…, k-1.

Chu trình: là một băng chuyền khép kín.

3. DÂY CHUYỀN, BĂNG CHUYỀN VÀ CHU TRÌNH

Ví dụ:



[1,2]-[2,4]-[5,4] là dây chuyền.

[1,3]-[3,2]-[2,4] là băng chuyền.

[3,2]-[2,4]-[4,3] là chu trình

3. DÂY CHUYỀN, BĂNG CHUYỀN VÀ CHU TRÌNH

Dây chuyền (băng chuyền) cơ bản là những dây chuyền (băng chuyền) mà mỗi đỉnh chỉ xuất hiện nhiều nhất một lần.

Băng chuyền cơ bản chứa tất cả các đỉnh của đồ thị được gọi là băng chuyền Hamilton.

Chu trình cơ bản chứa tất cả các cạnh của đồ thị được gọi là chu trình Euler.

Khuyên được coi là chu trình có độ dài 1.

4. ĐỒ THỊ CÓ HƯỚNG LIÊN THÔNG

Cho G = [V, E], khi đó:

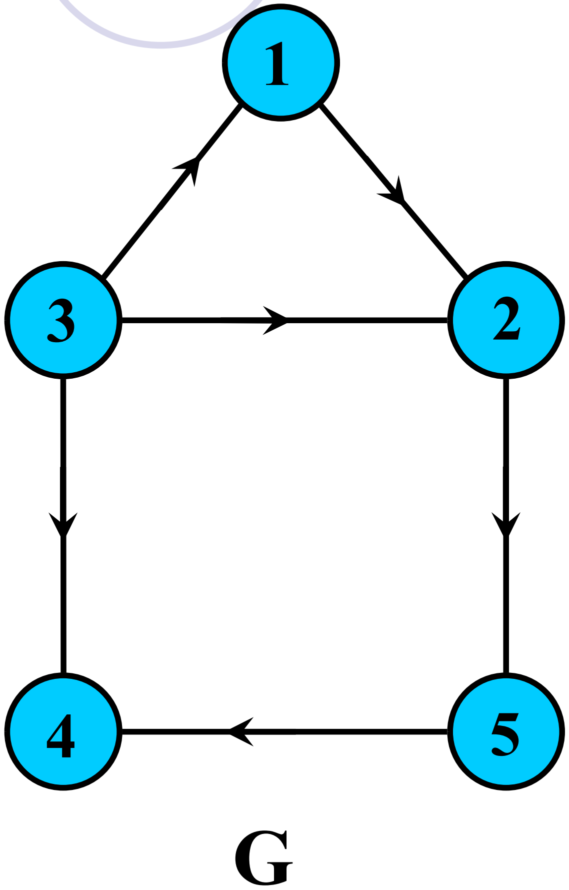
+ 2 đỉnh v và w được gọi là liên thông yếu nếu tồn tại dây truyền nối chúng.

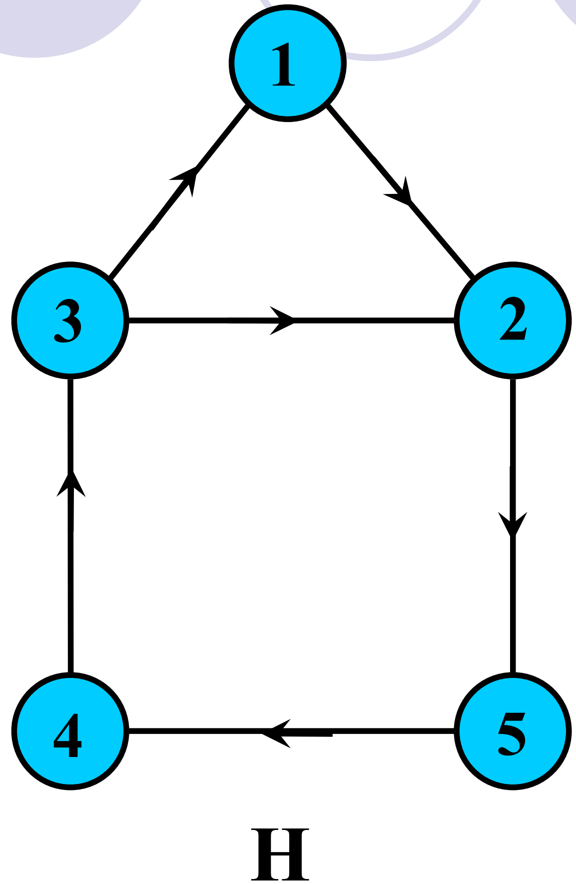
+ G là ***liên thông yếu*** nếu hai đỉnh bất kỳ đều liên thông yếu.

+ G là ***liên thông mạnh*** nếu hai đỉnh bất kỳ được nối với nhau bởi một băng chuyền.

4. ĐỒ THỊ CÓ HƯỚNG LIÊN THÔNG

Ví dụ:





G là liên thông yếu

H là liên thông mạnh

5. ĐỒ THỊ PHẢN CHU TRÌNH

Đồ thị phản chu trình: là một đồ thị có hướng G không chứa chu trình.

Định lý 1: trong mỗi đồ thị phản chu trình luôn tồn tại ít nhất một đỉnh nguồn và một đỉnh hạ lưu.

Định lý 2: đồ thị có hướng G = [V, E] với n đỉnh là đồ thị phản chu trình khi và chỉ khi chúng ta có thể đánh số các đỉnh của đồ thị bởi 1, 2, …, n sao cho mỗi cạnh [i, j] thoả mãn quan hệ i < j.

5. ĐỒ THỊ PHẢN CHU TRÌNH

Ví dụ: