

# Chương 2. CÂY

## Phần I. Hướng dẫn sử dụng Maple

Để thực hành các bài toán liên quan tới đồ thị và cây chúng ta sử dụng gói lệnh [GraphTheory](#). Để gọi gói lệnh này ta dùng

```
> with(GraphTheory);  
[AcyclicPolynomial, AddArc, AddEdge, AddVertex, AdjacencyMatrix, AllPairsDistance, ...]
```

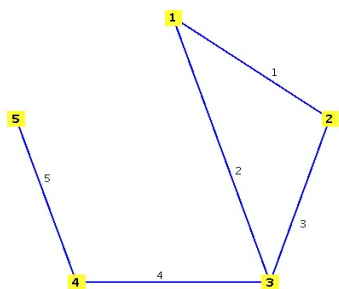
### 2.1 Đồ thị có trọng số

- **Graph(E)**: Tạo ra đồ thị có tập cạnh (cung) **E** và có trọng số. Nếu  $E = \{\{i, j\}, w, \dots\}$  thì đồ thị có được là vô hướng, trong đó cạnh  $ij$  có trọng số là  $w$ . Nếu  $E = \{[i, j], w, \dots\}$  thì đồ thị có được là có hướng, trong đó cung  $ij$  có trọng số là  $w$ .
- **Graph(V, E)**: tương tự như **Graph(E)** trong đó danh sách đỉnh là **V**.
- **Edges(G, weights)**: Danh sách các cạnh của đồ thị **G** kèm theo trọng số
- **WeightMatrix(G)**: Ma trận trọng số của đồ thị **G**.
- **Graph(W, weighted)**: Tạo ra đồ thị vô hướng có trọng số từ ma trận trọng số **W**
- **Graph(directed, W, weighted)**: Tạo ra đồ thị có hướng có trọng số từ ma trận trọng số **W**

```
> with(GraphTheory);  
> E := { [1, 2], 1, [1, 3], 2, [2, 3], 3, [3, 4], 4, [4, 5], 5 };  
G := Graph(E);
```

G := 'Graph 1: an undirected weighted graph with 5 vertices and 5 edge(s)'

```
> DrawGraph(G);
```

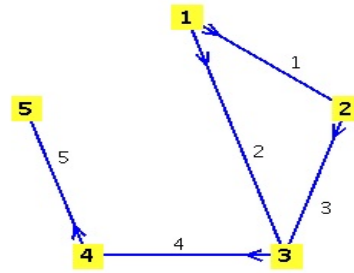


```
> V := [1, 2, 3, 4, 5, 6]:  
E := { [1, 2], 1, [1, 3], 2, [2, 3], 3, [3, 4], 4, [4, 5], 5 }:  
> H := Graph(V, E);
```

H := 'Graph 2: a directed weighted graph with 6 vertices and 5 arc(s)'

```
> DrawGraph(H);
```

6



> Edges(H, weights);

[[1, 2], 1], [[1, 3], 2], [[2, 3], 3], [[3, 4], 4], [[4, 5], 5]

> WeightMatrix(H);

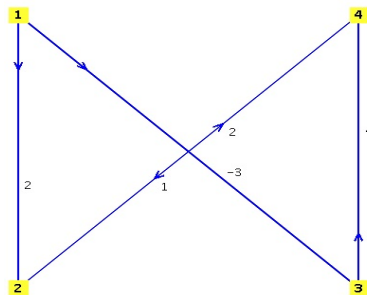
$$\begin{bmatrix} 0 & 1 & 2 & 0 & 0 & 0 \\ 0 & 0 & 3 & 0 & 0 & 0 \\ 0 & 0 & 0 & 4 & 0 & 0 \\ 0 & 0 & 0 & 0 & 5 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

> W := Matrix([ [0, 2, -3, 0], [0, 0, 0, 2], [0, 0, 0, 4], [0, 1, 0, 0] ]):

G := Graph(directed, W, weighted);

G := Graph 3: a directed weighted graph with 4 vertices and 5 arc(s)

> DrawGraph(G);



Ngoài ra để tạo ngẫu nhiên một đồ thị có trọng số, ta sử dụng thêm gói lệnh **RandomGraphs**. Nghĩa là gọi > **with(RandomGraphs);**

- **RandomGraph(n, m, weights = i..j):** Tạo ra đồ thị vô hướng **liên thông** có trọng số có **n** đỉnh, **m** cạnh và trọng lượng mỗi cạnh có giá trị trong đoạn **[i, j]**
- **RandomGraph(n, m, connected, weights = i..j):** Tạo ra đồ thị vô hướng có trọng số có **n** đỉnh, **m** cạnh và trọng lượng mỗi cạnh có giá trị trong đoạn **[i, j]**
- **RandomGraph(n, m, directed, weights = i..j):** Tạo ra đồ thị **có hướng** có trọng số có **n** đỉnh, **m** cạnh và trọng lượng mỗi cạnh có giá trị trong đoạn **[i, j]**

```
> with(RandomGraphs);
```

```
[AssignEdgeWeights, RandomBipartiteGraph, RandomDigraph, RandomGraph, ...]
```

```
> G := RandomGraph(6, 7, weights = -10 .. 5):
```

```
DrawGraph(G);
```

```
> H := RandomGraph(10, 9, connected, weights = -15 .. 5):
```

```
DrawGraph(H);
```

```
> T := RandomGraph(5, 6, directed, weights = 8 .. 20):
```

```
DrawGraph(T);
```

## 2.2 Cây

- `RandomTree(n)`: Tạo ngẫu nhiên một cây có  $n$  đỉnh.
- `RandomTree(n, degree < k)`: Tạo ngẫu nhiên một cây có  $n$  đỉnh và bậc mỗi đỉnh nhỏ hơn  $k$ .
- `RandomTree(n, weights = i .. j)`: Tạo ngẫu nhiên một cây có trọng số có  $n$  đỉnh và trọng lượng mỗi cạnh có giá trị trong đoạn  $[i, j]$ .
- `RandomTree(n, degree < k, weights = i .. j)`: Tạo ngẫu nhiên một cây có trọng số có  $n$  đỉnh, bậc mỗi đỉnh nhỏ hơn  $k$  và trọng lượng mỗi cạnh có giá trị trong đoạn  $[i, j]$ .

```
> with(GraphTheory):
```

```
with(RandomGraphs):
```

```
> T1 := RandomTree(20):
```

```
DrawGraph(T1);
```

```
> T2 := RandomTree(30, degree < 4):
```

```
DrawGraph(T2);
```

```
> T3 := RandomTree(40, weights = 1 .. 10):
```

```
DrawGraph(T3);
```

```
> T4 := RandomTree(50, degree < 5, weights = 2 .. 5):
```

```
DrawGraph(T4);
```

Một số lệnh liên quan đồ thị và cây

- `IsTree(G)`: Kiểm tra  $G$  có phải là cây hay không?
- `SpanningTree(G)`: Tìm một cây khung của đồ thị  $G$ .
- `NumberOfSpanningTrees(G)`: Số lượng cây khung của đồ thị  $G$ .
- `MinimalSpanningTree(G)`: Tìm một cây khung ngắn nhất của đồ thị có trọng số  $G$ .

> G := Graph({{1, 2}, {1, 3}}):

IsTree(G);

*true*

> H := Graph({{1, 2}, {1, 3}, {2, 3}}):

IsTree(H);

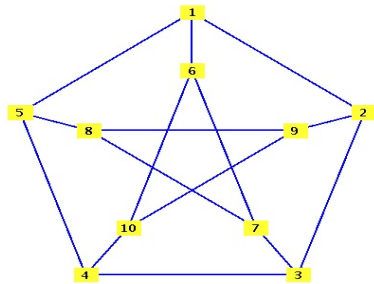
*false*

> with(SpecialGraphs); # gói lệnh chứa một số đồ thị đặc biệt

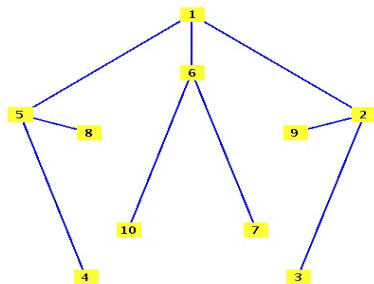
> P := PetersenGraph();

> T := SpanningTree(P);

> DrawGraph(P);



> DrawGraph(T);



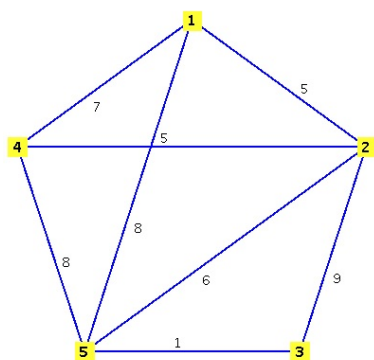
> NumberOfSpanningTrees(P);

2000

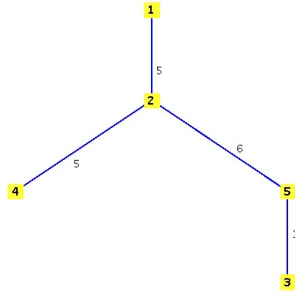
> W := Matrix([[0, 5, 0, 7, 8], [5, 0, 9, 5, 6], [0, 9, 0, 0, 1], [7, 5, 0, 0, 8], [8, 6, 1, 8, 0]]):

> G := Graph(W):

DrawGraph(G):



```
> T := MinimalSpanningTree(G);  
DrawGraph(T);
```



## ► Bài tập thực hành

**Bài 1.** Cho ma trận kề của đồ thị đơn vô hướng  $G$ . Hãy viết chương trình kiểm tra  $G$  có là cây không?

**Bài 2.** Cho ma trận kề của đồ thị đơn vô hướng liên thông  $G$ . Hãy viết chương trình tìm cây khung của  $G$  bằng

a) thuật toán BFS

b) thuật toán DFS

**Bài 3.** Cho danh sách các cạnh và trọng lượng tương ứng của chúng của một đồ thị  $G$ . Hãy viết chương trình tìm ma trận trọng số của  $G$ .

**Bài 4.** Cho ma trận trọng số của đồ thị vô hướng liên thông  $G$ . Hãy tìm cây khung ngắn nhất và cây khung dài nhất của  $G$  bằng

a) thuật toán Kruskal

b) thuật toán Prim

**Bài 5.** Cho biểu thức số học dưới dạng tiền tố (hậu tố, trung tố). Hãy viết chương trình tính giá trị của nó.

## Phần II. Bài tập

**2.1** Cho các cây  $T_i = (X_i, E_i)$  với  $n_i = |X_i|$  và  $m_i = |E_i|$  ( $i = 1, 2$ ). Tính  $n_1, n_2$  và  $m_2$  nếu biết  $m_1 = 17$  và  $n_2 = 2n_1$ .

**2.2** Cho  $G$  là một rừng có 7 cây và 40 cạnh. Tìm số đỉnh của  $G$ .

**2.3** Cho  $G$  là một cây có một đỉnh bậc 3, hai đỉnh bậc 4, một đỉnh bậc 5 và các đỉnh còn lại có bậc  $\leq 2$ . Tính số đỉnh treo của  $G$ .

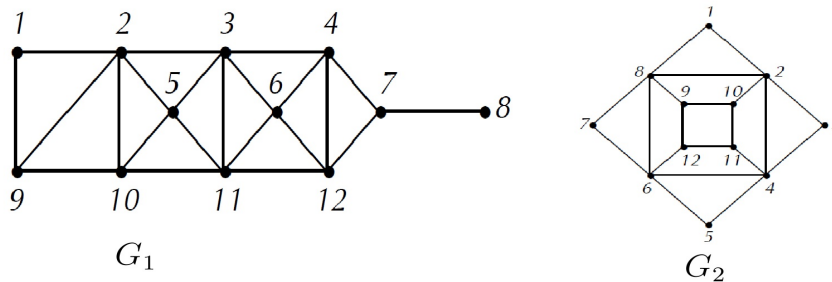
**2.4** Cho  $T = (X, E)$  là một cây với  $X = \{1, 2, \dots, n\}$ . Chứng minh rằng số đỉnh treo của  $T$  là

$$2 + \sum_{i=1d(i) \geq 3}^n (\deg(i) - 2).$$

**2.5** Cho  $G$  là một đồ thị vô hướng gồm  $n$  đỉnh,  $m$  cạnh và  $p$  thành phần liên thông ( $m \geq 0, n \geq 1, p \geq 1$ ). Chứng minh rằng

- $m \geq n - p$ .
- $G$  là một rừng nếu và chỉ nếu  $m - n + p = 0$ .

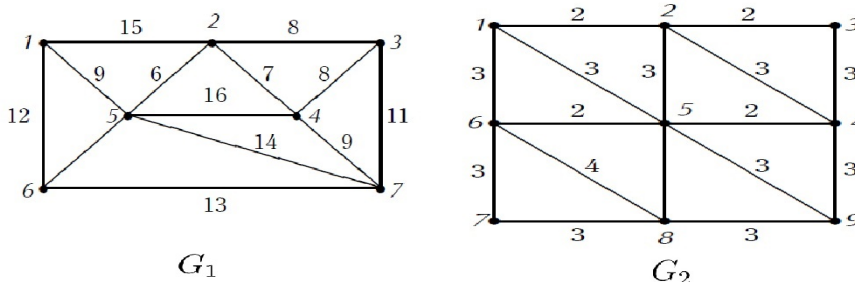
**2.6** Hãy tìm cây khung của hai đồ thị sau bằng thuật toán BFS và DFS:



**2.7**

- Chứng minh mỗi cây là một đồ thị lưỡng phân.
- Cho  $G$  là đồ thị vô hướng. Chứng minh  $G$  là đồ thị lưỡng phân khi và chỉ khi  $G$  không có chu trình nào hoặc mọi chu trình của  $G$  đều có độ dài chẵn.

**2.8** Cho hai đồ thị sau:



Đối với mỗi đồ thị hãy dùng thuật toán Kruskal và Prim để

- a) tìm cây khung ngắn nhất.
- b) tìm cây khung ngắn nhất chứa cạnh 34.
- c) tìm cây khung dài nhất nhất.
- d) tìm cây khung dài nhất có chứa cạnh 67 và không chứa cạnh 34.

**2.9** Vẽ một cây nhị phân đủ có chiều cao  $h = 3$ , có 4 đỉnh trong và 5 lá.

**2.10** Cho  $T$  là cây  $k$ -phân đủ có  $m$  đỉnh trong. Chứng minh rằng số đỉnh của  $T$  là  $km + 1$ .

**2.11** Cho  $T = (X, E)$  là một cây tam phân đủ có 34 đỉnh trong. Tính số cạnh và số lá của  $T$ .

**2.12** Cho  $T = (X, E)$  là một cây ngũ phân đủ có 817 lá. Hỏi  $T$  có bao nhiêu đỉnh trong?

**2.13** Cho  $T$  là một cây tứ phân đủ có chiều cao 8. Hỏi  $T$  có nhiều nhất là bao nhiêu đỉnh trong? Còn nếu  $T$  là  $k$ -phân đủ có chiều cao  $h$  thì số đỉnh trong tối đa của  $T$  là bao nhiêu?

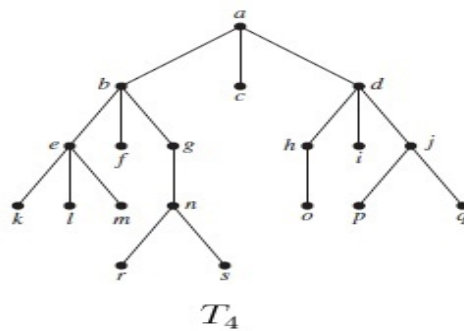
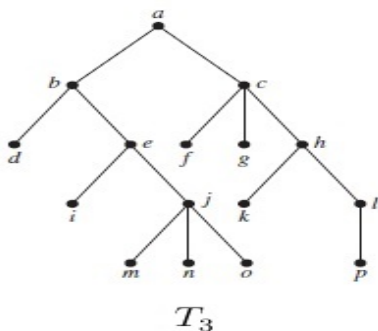
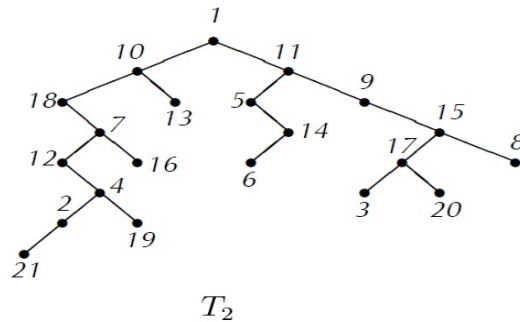
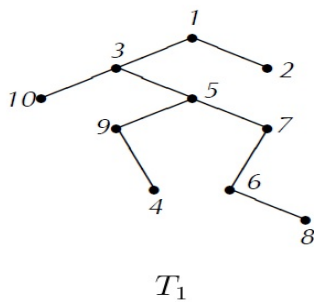
**2.14** Một cây  $k$ -phân đủ có chiều cao  $h$  được gọi là cây  $k$ -phân đầy nếu tất cả các lá của nó đều ở mức  $h$ . Tìm số lá của cây nhị phân đầy nếu

- a)  $h = 3$ .
- b)  $h = 7$ .
- c)  $h = 12$ .

**2.15** Cho  $T$  là cây nhị phân đầy. Tính số đỉnh trong và số cạnh của  $T$  biết chiều cao của  $T$  là  $h = 5$ .

**2.16** Cho  $T$  là cây  $k$ -phân đầy có chiều cao 7 và 279 936 lá. Hỏi  $T$  có bao nhiêu đỉnh trong?

**2.17** Cho các cây sau



Hãy liệt kê các đỉnh của các cây khi dùng các phép duyệt tiền thứ tự, hậu thứ tự và trung thứ tự.

**2.18** Viết các biểu thức sau đây bằng ký pháp Balan và ký pháp Balan ngược rồi vẽ cây nhị phân của các biểu thức tương ứng.

$$\text{a)} \quad \frac{w+x-y}{\pi x^3}$$

$$\text{c)} \quad (((a+b)*c+d)*e) - ((a+b)*c+d).$$

$$\text{b)} \quad (n^n)^n(mn-q)$$

**2.19** Tính giá trị biểu thức được viết bằng ký pháp Balan sau

$$\text{a)} \quad +4/\ast 2\ 3+1-9\wedge 2\ 3$$

$$\text{b)} \quad \wedge 2-4\ 2+2\ast 2\ 4$$

Lưu ý:  $\wedge$  là phép toán lấy lũy thừa.

**2.20** Tính giá trị biểu thức được viết bằng ký pháp Balan ngược sau

$$\text{a)} \quad 1\ 2+3\ 4\ast 1\ 1/-2\ast$$

$$\text{b)} \quad 1\ 2\ 1\ 2\ast +\ast 4\ast$$

$$\text{c)} \quad 1\ 4\ 2\ 3\ 4\ast -+\ast$$