Toán rời rạc 2

Discrete mathematics 2

Bài 5: Cây và Cây khung của đồ thị

Trees and Spanning Trees

Link(s) download slide bài giảng

3. Tìm kiếm - duyệt đồ thị (https://bit.ly/3cq8TNQ)

   Graph Traversal

4. Đồ thị Euler và Hamilton (https://bit.ly/3cmZ0QS)

   Eulerian & Hamiltonian Graphs

5. Cây và Cây khung của đồ thị (https://bit.ly/2Ik1LVe)

   Trees and Spanning Trees

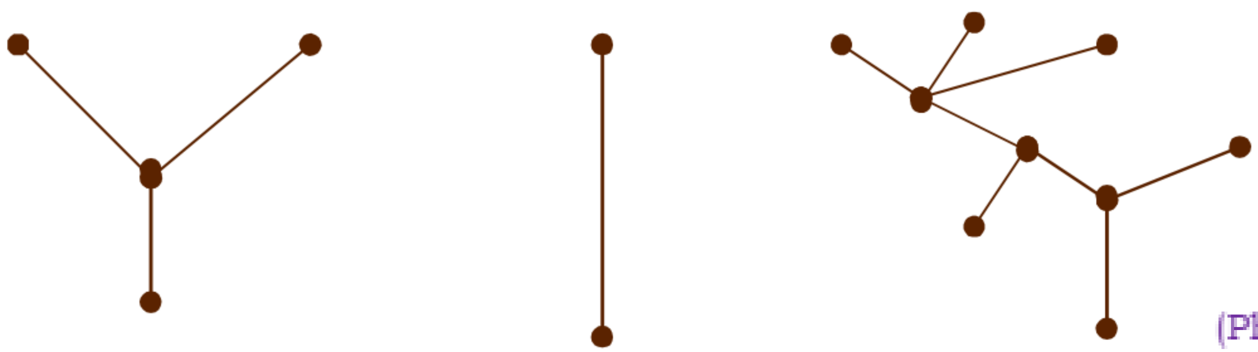
... (tiếp tục cập nhật)

Nội dung Bài 5

1. Cây và các tính chất của cây
2. Cây khung của đồ thị
3. Bài toán cây khung nhỏ nhất

Định nghĩa và ví dụ

- Định nghĩa 1: Ta gọi cây là một đồ thị vô hướng, liên thông, không có chu trình.
- Định nghĩa 2: Ta gọi rừng là một đồ thị vô hướng, không có chu trình.
- Như vậy rừng là một đồ thị mà mỗi thành phần liên thông của nó là một cây.
- Ví dụ dưới đây: một rừng có 3 cây



(Phương ND, 2013)

Các tính chất của cây

- Định lý: Giả sử $T = < V, E >$ là đồ thị vô hướng 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 liên thông và có đúng $n - 1$ cạnh.
  4. T liên thông và mỗi cạnh của nó đều là cầu.
  5. Giữa hai đỉnh bất kỳ của T được nối với nhau bởi đúng một đường đi đơn.
  6. T không chứa chu trình nhưng hễ cứ thêm vào nó một cạnh ta thu được đúng một chu trình.

  - Chứng minh:

    Theo sơ đồ (1) $\Rightarrow$ (2) $\Rightarrow$ (3) $\Rightarrow$ (4) $\Rightarrow$ (5) $\Rightarrow$ (6) $\Rightarrow$ (1)

Nội dung Bài 5

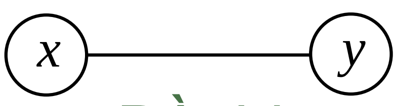
1. Cây và các tính chất của cây

2. Cây khung của đồ thị

3. Bài toán cây khung nhỏ nhất

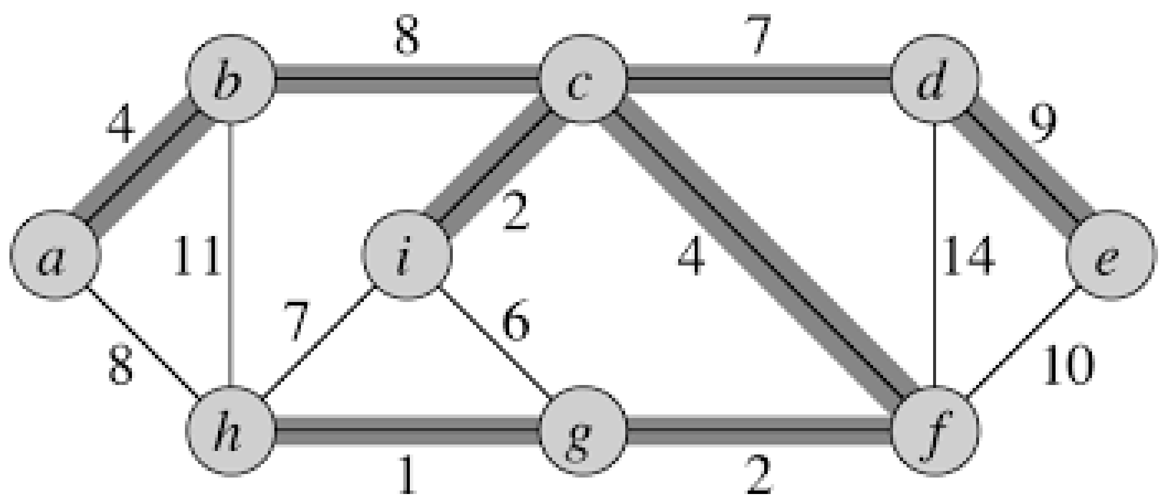
Định nghĩa và ví dụ (1/2)

- Định nghĩa 3: 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:
  1. T là một cây.
  2. Tập đỉnh của T bằng tập đỉnh của G.
- Ví dụ:

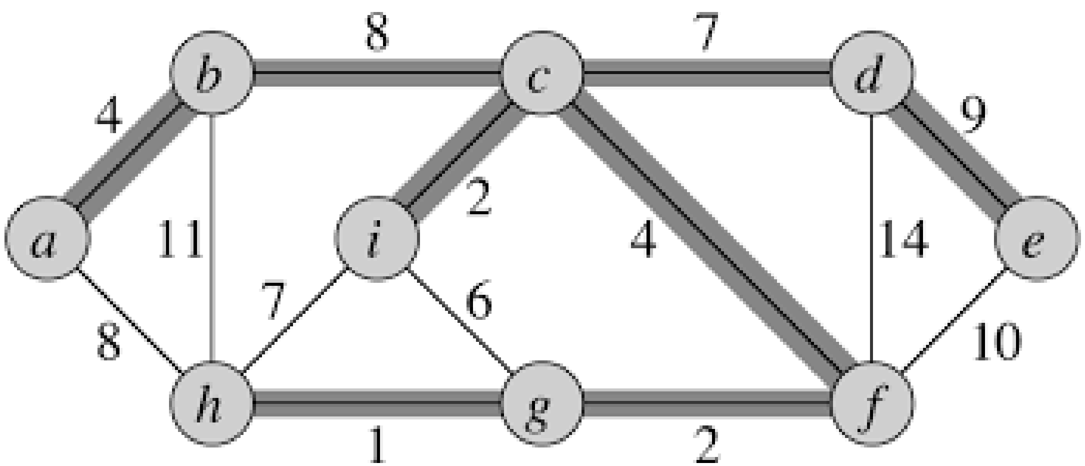


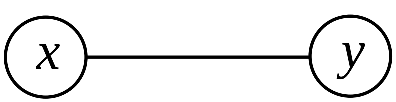
Đồ thị

Cây bao trùm



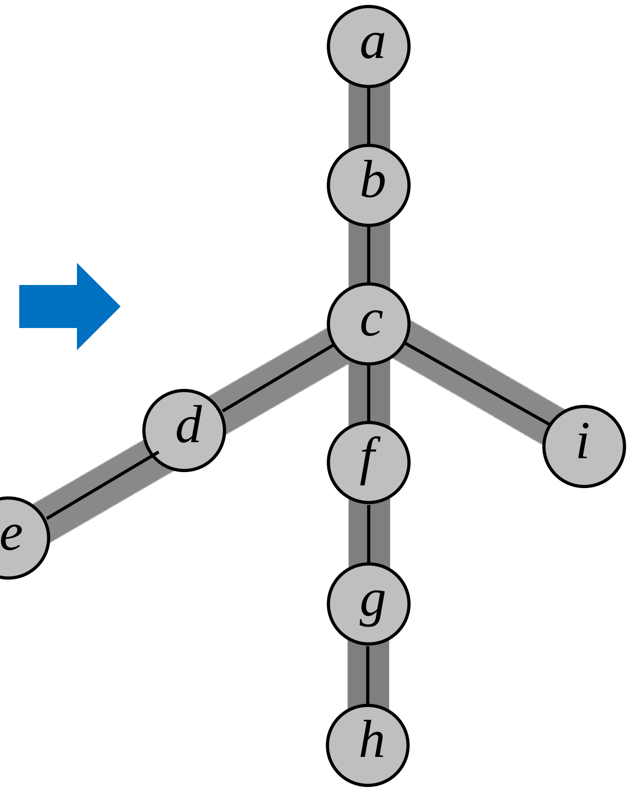
Định nghĩa và ví dụ (2/2)





Đồ thị

Cây bao trùm



Cây bao trùm – cây khung của đồ thị

Định lý số 1

- Phát biểu:

Một đơn đồ thị là liên thông khi và chỉ khi nó có cây bao trùm.

- Chứng minh:

Rosen, K.H., 2018. *Discrete Mathematics & Applications.* McGraw-Hill.

Xây dựng cây khung của đồ thị

- **Bài toán:**
  Cho đồ thị vô hướng $G = < V, E >$.
  Hãy xây dựng một cây khung của đồ thị bắt đầu tại đỉnh $u \in V$.

- **Cách làm:**
  - Sử dụng thuật toán duyệt DFS hoặc BFS;
  - Mỗi khi ta đến được đỉnh v (tức là $chuaxet[v] = true$) từ đỉnh u thì cạnh (u, v) được kết nạp vào cây khung.

Tạo cây khung của đồ thị với DFS (1/5)

- Thuật toán tạo 1 cây khung từ một đỉnh $\boldsymbol{u}$:

Tree-DFS(u){

$chuaxet[u] = false$; *// đánh dấu đỉnh u đã duyệt*

for($v \in Ke(u)$){

if($chuaxet[v]$){ *// nếu v chưa được duyệt*

$T = T \cup \{u, v\}$; *// hợp cạnh (u, v) vào cây khung T*

Tree-DFS(v); *// duyệt theo chiều sâu từ v*

}

}

}

Tạo cây khung của đồ thị với DFS (2/5)

- Chương trình:

root = <*đỉnh u bất kỳ* ∈ V>; *// Lấy một đỉnh bất kỳ làm gốc*

Tree-Graph-DFS(*root*){

  for($u \in V$)

    $chuaxet[u] = true$; *// Khởi tạo mọi đỉnh: chưa xét*

  $T = \emptyset$; *// Cây ban đầu chưa có cạnh nào*

  Tree-DFS(root); *// Gọi hàm tạo cây khung từ 1 đỉnh*

  if($T < n - 1$)

    <đồ thị không liên thông>;

  else

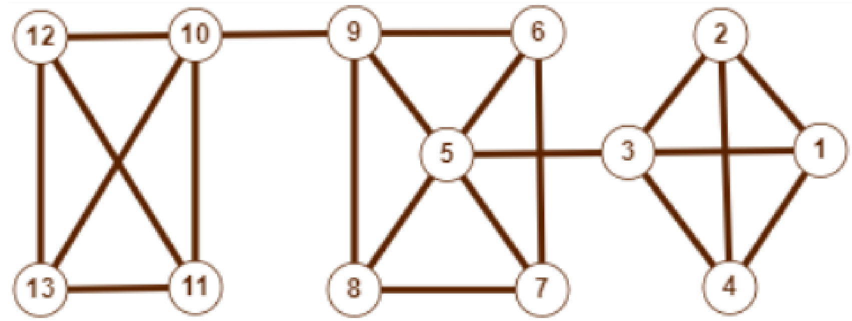
    <ghi nhận tập cạnh của cây khung T>;

}

Kiểm nghiệm thuật toán (3/5)

Cho đồ thị vô hướng được biểu diễn bằng ma trận kề như hình bên.

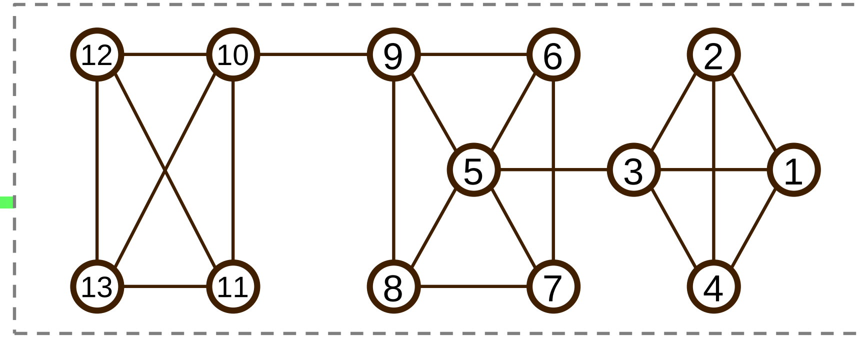
- Áp dụng thuật toán xây dựng cây khung của đồ thị sử dụng DFS cho đồ thị trên bắt đầu từ đỉnh $u = 1$.



$$
\begin{vmatrix}
0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0
\end{vmatrix}
$$

(Phương ND, 2013)

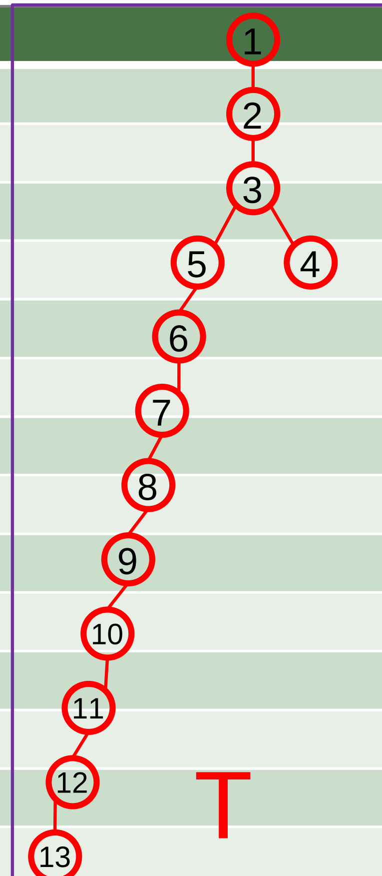
Kiểm nghiệm (4/5)



- Gọi: Tree-Graph-DFS(1)

Đã xét   Các đỉnh kề

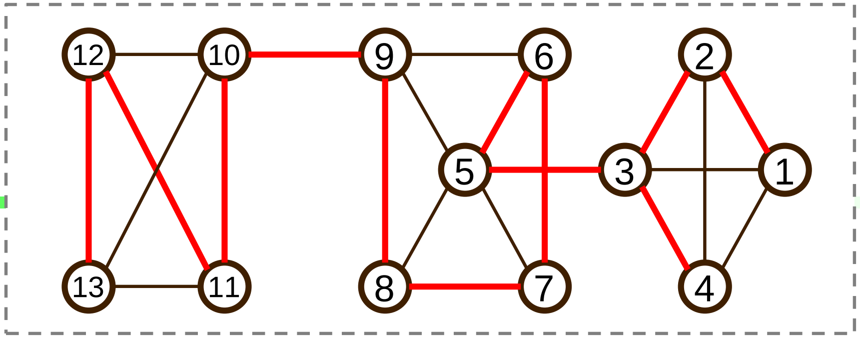
#	Ngăn xếp	T
0	1	T = ∅
1	1, 2	T = T ∪ {(1, 2)}
2	1, 2, 3	T = T ∪ {(2, 3)}
3	1, 2, 3, 4	T = T ∪ {(3, 4)}
4	1, 2, 3	
5	1, 2, 3, 5	T = T ∪ {(3, 5)}
6	1, 2, 3, 5, 6	T = T ∪ {(5, 6)}
7	1, 2, 3, 5, 6, 7	T = T ∪ {(6, 7)}
8	1, 2, 3, 5, 6, 7, 8	T = T ∪ {(7, 8)}
9	1, 2, 3, 5, 6, 7, 8, 9	T = T ∪ {(8, 9)}
10	1, 2, 3, 5, 6, 7, 8, 9, 10	T = T ∪ {(9, 10)}
11	1, 2, 3, 5, 6, 7, 8, 9, 10, 11	T = T ∪ {(10, 11)}
12	1, 2, 3, 5, 6, 7, 8, 9, 10, 11, 12	T = T ∪ {(11, 12)}
13	1, 2, 3, 5, 6, 7, 8, 9, 10, 11, 12, 13	T = T ∪ {(12, 13)}



Không thêm được cạnh nào nữa vào T

T = {(1, 2), (2, 3), (3, 4), (3, 5), (5, 6), (6, 7), (7, 8), (8, 9), (9, 10), (10, 11), (11, 12), (12, 13)}

Kiểm nghiệm (5/5)



Cây bao trùm

- Gọi: Tree-Graph-DFS(1)

```
So dinh do thi: 13
Ma tran ke:
 0   1   1   1   0   0   0   0   0   0   0   0   0
 1   0   1   1   0   0   0   0   0   0   0   0   0
 1   1   0   1   1   0   0   0   0   0   0   0   0
 1   1   1   0   0   0   0   0   0   0   0   0   0
 0   0   1   0   0   1   1   1   1   0   0   0   0
 0   0   0   0   1   0   1   0   1   0   0   0   0
 0   0   0   0   1   1   0   1   0   0   0   0   0
 0   0   0   0   1   0   1   0   1   0   0   0   0
 0   0   0   0   1   1   0   1   0   1   0   0   0
 0   0   0   0   0   0   0   0   1   0   1   1   1
 0   0   0   0   0   0   0   0   0   1   0   1   1
 0   0   0   0   0   0   0   0   0   1   1   0   1
 0   0   0   0   0   0   0   0   0   1   1   1   0
```

```
n = 13, sc = 12
Canh 1:   1  2
Canh 2:   2  3
Canh 3:   3  4
Canh 4:   3  5
Canh 5:   5  6
Canh 6:   6  7
Canh 7:   7  8
Canh 8:   8  9
Canh 9:   9 10
Canh 10: 10 11
Canh 11: 11 12
Canh 12: 12 13
```

Tạo cây khung của đồ thị với BFS (1/4)

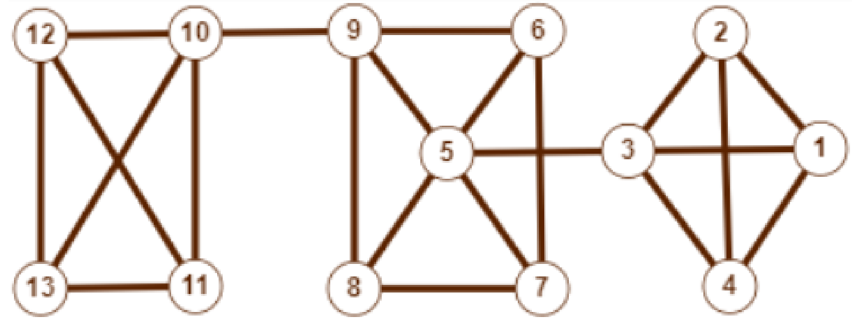
- Thuật toán tạo cây khung từ một đỉnh $\boldsymbol{u}$:

```
Tree-BFS(u){
    Bước 1: Khởi tạo
    T = ∅; queue = ∅; push(queue, u); chuaxet[u] = false;
    Bước 2: Lặp
    while(queue ≠ ∅){
        s = pop(queue);
        for(t ∈ Ke(s)){
            if(chuaxet[t]){
                push(queue, t); T = T ∪ {s, t}; chuaxet[t] = false;
            }
        }
    }
    Bước 3: Trả lại kết quả
    if(T < n − 1)      <đồ thị không liên thông>;
    else               <ghi nhận tập cạnh của cây khung T>;
}
```

Kiểm nghiệm thuật toán (2/4)

Cho đồ thị vô hướng được biểu diễn bằng ma trận kề như hình bên.

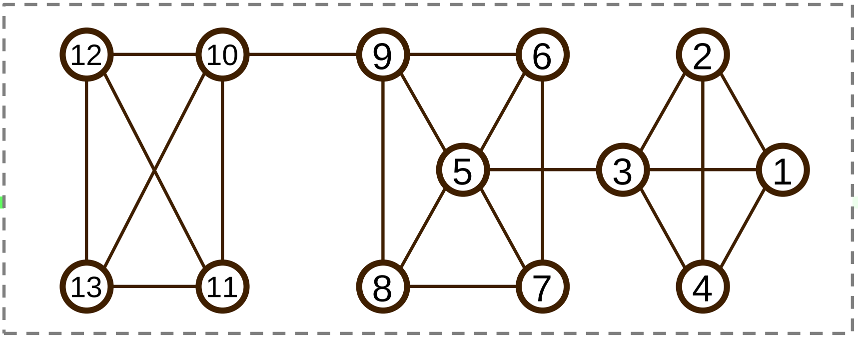
- Áp dụng thuật toán xây dựng cây khung của đồ thị sử dụng BFS cho đồ thị trên bắt đầu từ đỉnh $u = 1$.



$$\begin{vmatrix}
0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0
\end{vmatrix}$$

(Phương ND, 2013)

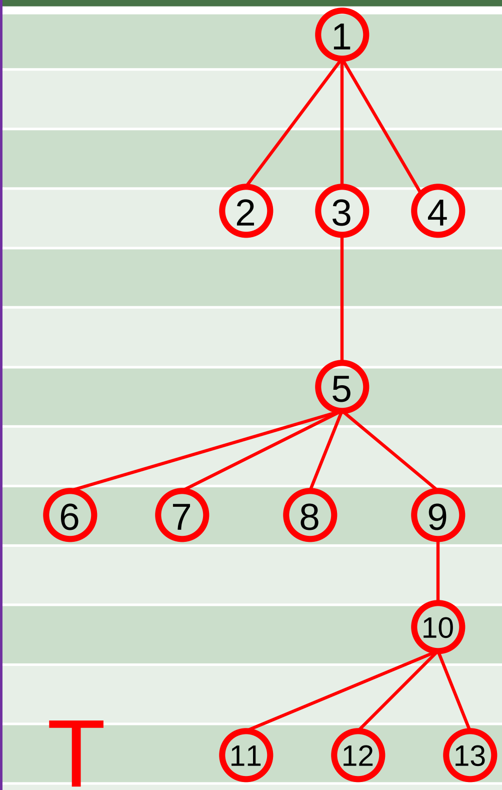
Kiểm nghiệm (3/4)



- Gọi: Tree-BFS(1)

○ Đã xét, trong queue ○ Các đỉnh kề
ⓥ Đã xét, đã lấy khỏi queue

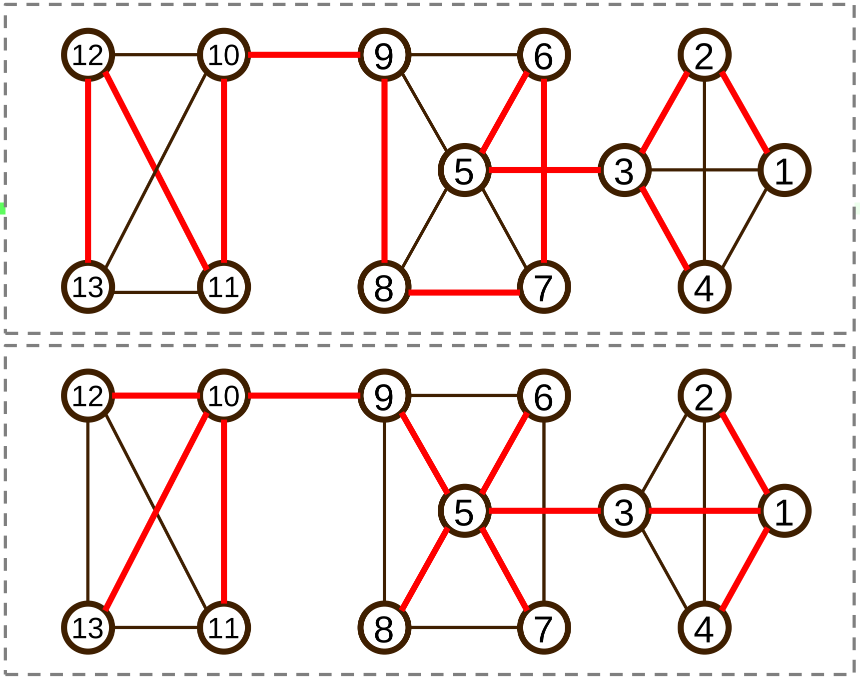
#	Queue	T
0	1	$T = \emptyset$
1	2, 3, 4	$T = T \cup \{(1, 2), (1, 3), (1, 4)\}$
2	3, 4	
3	4, 5	$T = T \cup \{(3, 5)\}$
4	5	
5	6, 7, 8, 9	$T = T \cup \{(5, 6), (5, 7), (5, 8), (5, 9)\}$
6	7, 8, 9	
7	8, 9	
8	9	
9	10	$T = T \cup \{(9, 10)\}$
10	11, 12, 13	$T = T \cup \{(10,11), (10, 12), (10, 13)\}$
11	12, 13	
12	13	
13	Ø	



$T = \{(1, 2), (1, 3), (1, 4), (3, 5), (5, 6), (5, 7), (5, 8), (5, 9), (9, 10), (10, 11), (10, 12), (10, 13)\}$

Kiểm nghiệm (4/4)

DFS



Cây bao trùm

BFS

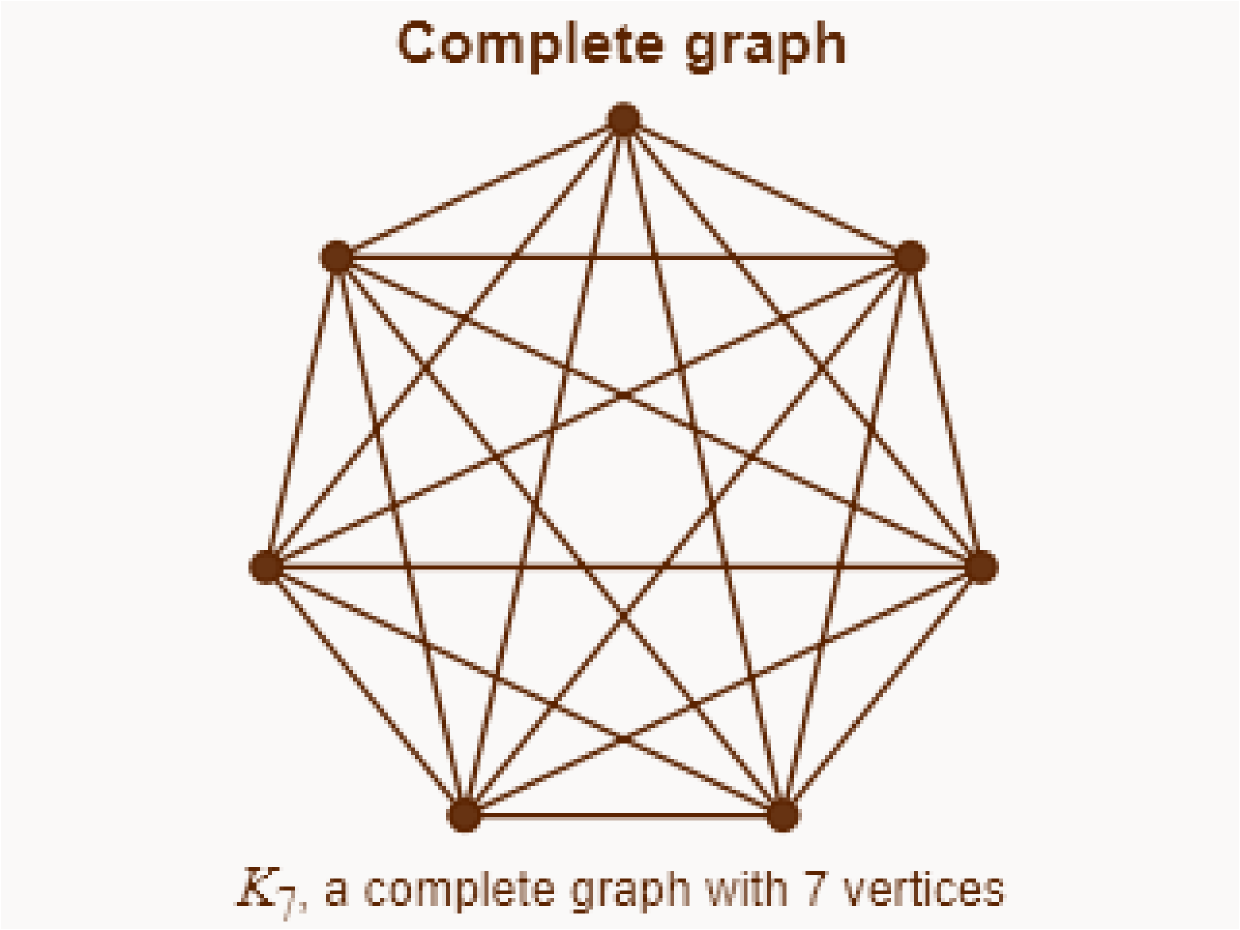
- Gọi: Tree-BFS(1)

```
So dinh do thi: 13
Ma tran ke:
 0  1  1  1  0  0  0  0  0  0  0  0  0
 1  0  1  1  0  0  0  0  0  0  0  0  0
 1  1  0  1  1  0  0  0  0  0  0  0  0
 1  1  1  0  0  0  0  0  0  0  0  0  0
 0  0  1  0  0  1  1  1  1  0  0  0  0
 0  0  0  0  1  0  1  0  1  0  0  0  0
 0  0  0  0  1  1  0  1  0  0  0  0  0
 0  0  0  0  1  0  1  0  1  0  0  0  0
 0  0  0  0  1  1  0  1  0  1  0  0  0
 0  0  0  0  0  0  0  0  1  0  1  1  1
 0  0  0  0  0  0  0  0  0  1  0  1  1
 0  0  0  0  0  0  0  0  0  1  1  0  1
 0  0  0  0  0  0  0  0  0  1  1  1  0
```

```
Canh 1:   1  2
Canh 2:   1  3
Canh 3:   1  4
Canh 4:   3  5
Canh 5:   5  6
Canh 6:   5  7
Canh 7:   5  8
Canh 8:   5  9
Canh 9:   9 10
Canh 10: 10 11
Canh 11: 10 12
Canh 12: 10 13
```

Số cây khung của đồ thị đầy đủ K_n

- Số cây khung của đồ thị đầy đủ n đỉnh K_n là $n^{(n-2)}$



Nội dung Bài 5

1. Cây và các tính chất của cây

2. Cây khung của đồ thị

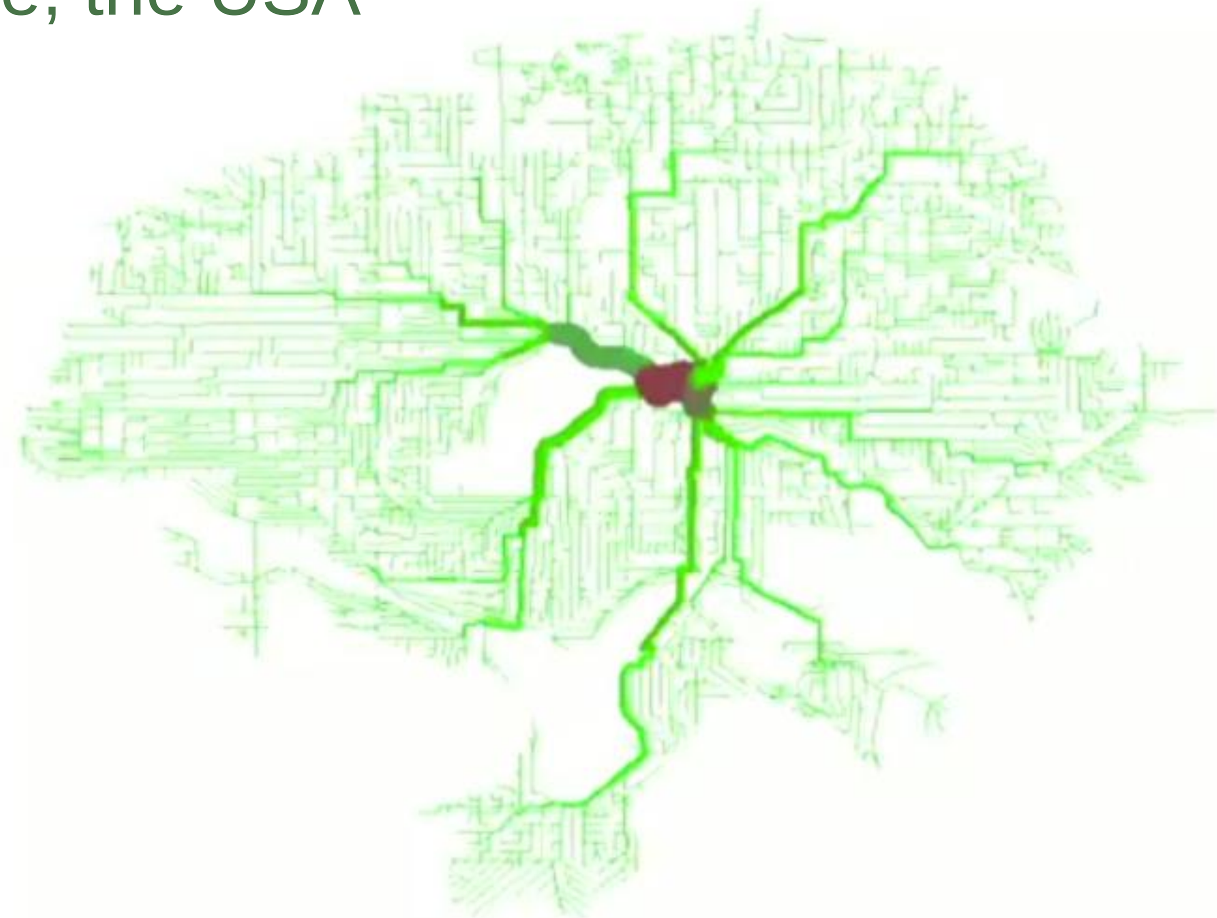
3. Bài toán cây khung nhỏ nhất

Một số ứng dụng (1/4)

1. Bài toán nối mạng máy tính:
   - Một mạng máy tính gồm n máy tính được đánh số từ 1, 2, . . . , n. Biết chi phí nối máy i với máy j là $c[i, j]$, $i, j = 1, 2, . . . , n$.
   - Hãy tìm cách nối mạng sao cho chi phí là nhỏ nhất.
2. Bài toán xây dựng hệ thống cáp
   - Giả sử ta muốn xây dựng một hệ thống cáp điện thoại nối n điểm của một mạng viễn thông sao cho điểm bất kỳ nào trong mạng đều có đường truyền tin tới các điểm khác. Biết chi phí xây dựng hệ thống cáp từ điểm i đến điểm j là $c[i, j]$.
   - Hãy tìm cách xây dựng hệ thống mạng cáp sao cho chi phí là nhỏ nhất.

Một số ứng dụng (2/4)

- Network design: MST of bycicle routes in North Seatle, the USA

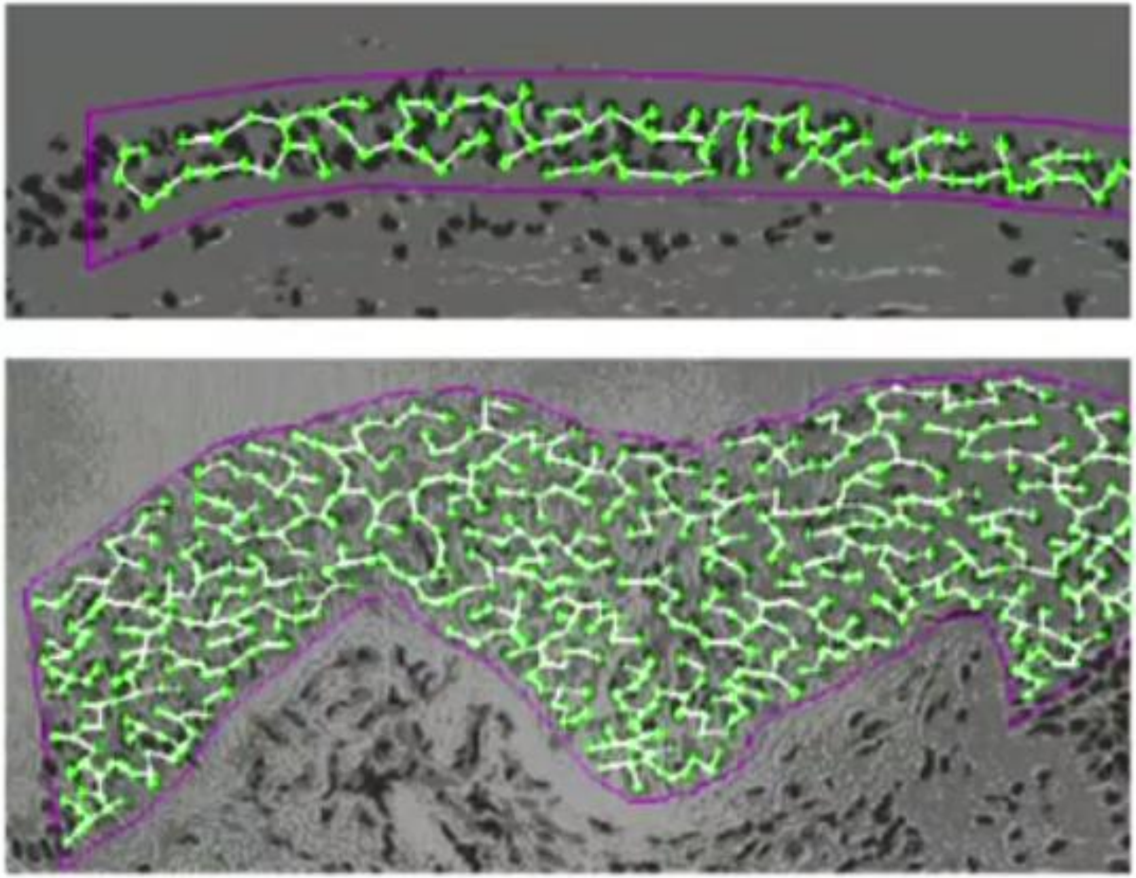


Một số ứng dụng (3/4)

- Models of nature: MST of random graph

Một số ứng dụng (4/4)

- MST describes arrangement of nuclei in the epithelium for cancer research.



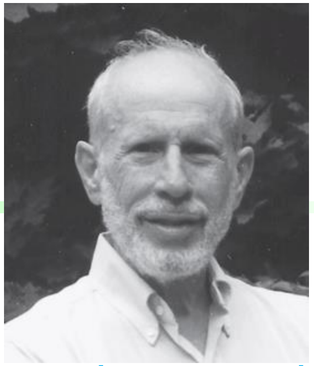
Phát biểu bài toán

- Cho $G = < V, E >$ là đồ thị vô hướng liên thông với tập đỉnh V và tập cạnh E.
  Mỗi cạnh e của đồ thị được gán với một số không âm $c(e)$ được gọi là độ dài cạnh.
  - Giả sử $H = < V, T >$ là một cây khung của đồ thị G.
    Ta gọi độ dài $c(H)$ của cây khung H là tổng độ dài các cạnh:

$$c(H) = \sum_{e \in T} c(e)$$

Bài toán: Trong số các cây khung của đồ thị hãy tìm cây khung có độ dài nhỏ nhất.

Thuật toán Kruskal (1/2)



Joseph Bernard KRUSKAL (1928–2010)

Thêm dần từng cạnh vào cây khung:

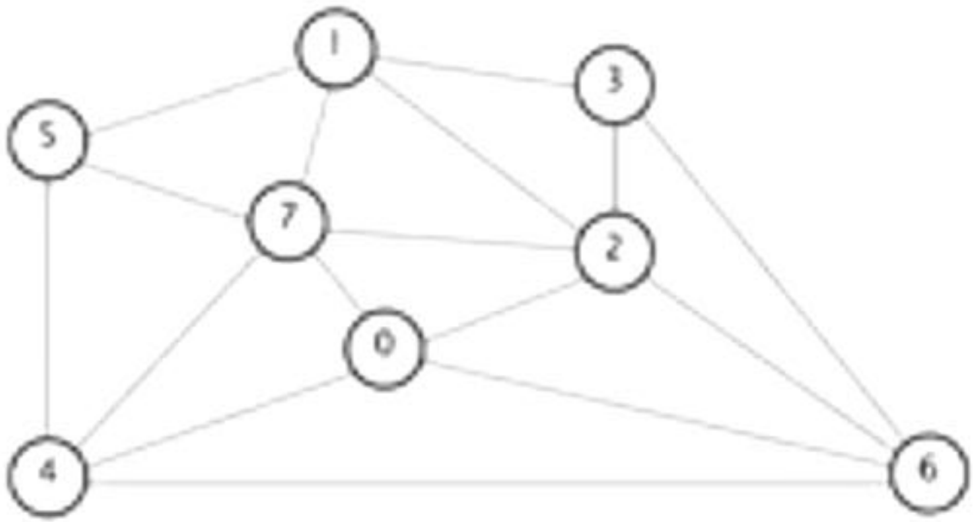
1. Mỗi bước chọn cạnh có trọng số nhỏ nhất chưa nằm trong cây khung.
2. Nếu việc thêm cạnh này vào cây khung không tạo thành chu trình thì thêm cạnh này vào.
3. Thuật toán dừng lại khi:

   Cây khung có đủ $(n - 1)$ cạnh.

Thuật toán Kruskal (2/2)

```
Kruskal( ){
    Bước 1 (khởi tạo):
    T = ∅;                          // Ban đầu tập cạnh cây khung là rỗng
    d(H) = 0;                       // Ban đầu độ dài cây khung là 0
    Bước 2 (sắp xếp):
    <sắp xếp các cạnh đồ thị theo thứ tự tăng dần của trọng số>;
    Bước 3 (lặp):
    while(|T| < n − 1 && E ≠ ∅ ){
        e = <Cạnh có độ dài nhỏ nhất>;
        E = E \ {e};                // Loại cạnh e ra khỏi tập cạnh
        if(T ∪ {e} không tạo nên chu trình){
            T = T ∪ {e};            // Đưa e vào cây khung
            d(H) = d(H) + d(e);     // Cập nhật độ dài cây khung
        }
    }
    Bước 4 (trả lại kết quả):
    if(|T| < n − 1) <Đồ thị không liên thông>;
    else            return (T, d(H));
}
```

Minh họa thuật toán Kruskal

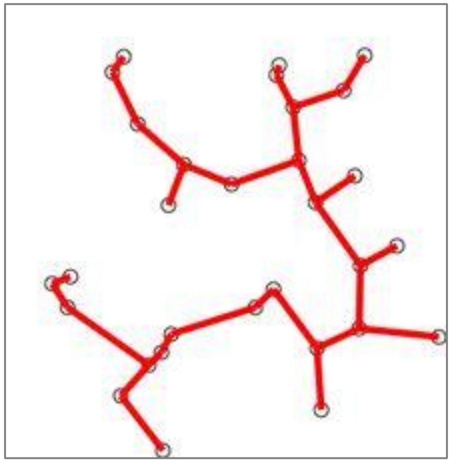


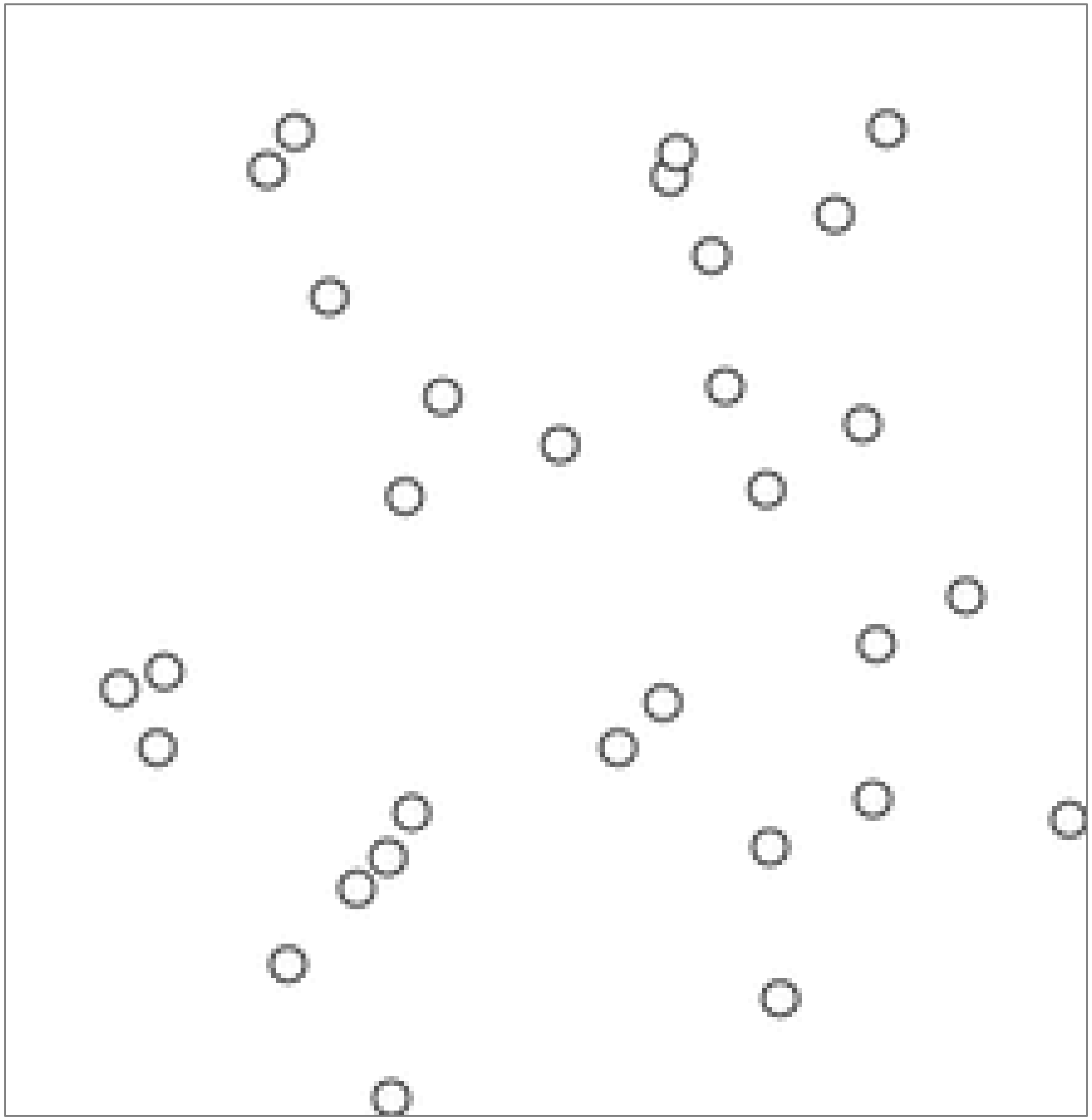
an edge-weighted graph

0-7	0.16
2-3	0.17
1-7	0.19
0-2	0.26
5-7	0.28
1-3	0.29
1-5	0.32
2-7	0.34
4-5	0.35
1-2	0.36
4-7	0.37
0-4	0.38
6-2	0.40
3-6	0.52
6-0	0.58
6-4	0.93

Quan sát thuật toán Kruskal

Cây bao trùm nhỏ nhất với trọng số là khoảng cách giữa 2 điểm

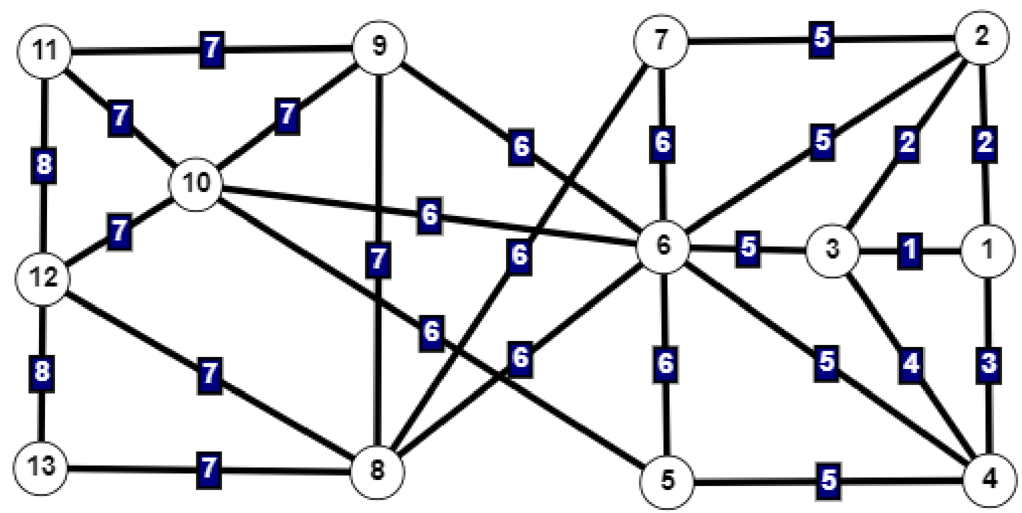




Kiểm nghiệm thuật toán Kruskal

Áp dụng thuật toán Kruskal tìm cây khung nhỏ nhất cho đồ thị được biểu diễn bằng ma trận trọng số như hình dưới?

$$\begin{vmatrix}
\infty & 2 & 1 & 3 & \infty & \infty & \infty & \infty & \infty & \infty & \infty & \infty & \infty \\
2 & \infty & 2 & \infty & \infty & 5 & 5 & \infty & \infty & \infty & \infty & \infty & \infty \\
1 & 2 & \infty & 4 & \infty & 5 & \infty & \infty & \infty & \infty & \infty & \infty & \infty \\
3 & \infty & 4 & \infty & 5 & 5 & \infty & \infty & \infty & \infty & \infty & \infty & \infty \\
\infty & \infty & \infty & 5 & \infty & 6 & \infty & \infty & \infty & 6 & \infty & \infty & \infty \\
\infty & 5 & 5 & 5 & 6 & \infty & 6 & 6 & 6 & 6 & \infty & \infty & \infty \\
\infty & 5 & \infty & \infty & \infty & 6 & \infty & 6 & \infty & \infty & \infty & \infty & \infty \\
\infty & \infty & \infty & \infty & \infty & 6 & 6 & \infty & 7 & \infty & \infty & 7 & 7 \\
\infty & \infty & \infty & \infty & \infty & 6 & \infty & 7 & \infty & 7 & 7 & \infty & \infty \\
\infty & \infty & \infty & \infty & 6 & 6 & \infty & \infty & 7 & \infty & 7 & 7 & \infty \\
\infty & \infty & \infty & \infty & \infty & \infty & \infty & \infty & 7 & 7 & \infty & 8 & \infty \\
\infty & \infty & \infty & \infty & \infty & \infty & \infty & 7 & \infty & 7 & 8 & \infty & 8 \\
\infty & \infty & \infty & \infty & \infty & \infty & \infty & 7 & \infty & \infty & \infty & 8 & \infty
\end{vmatrix}$$



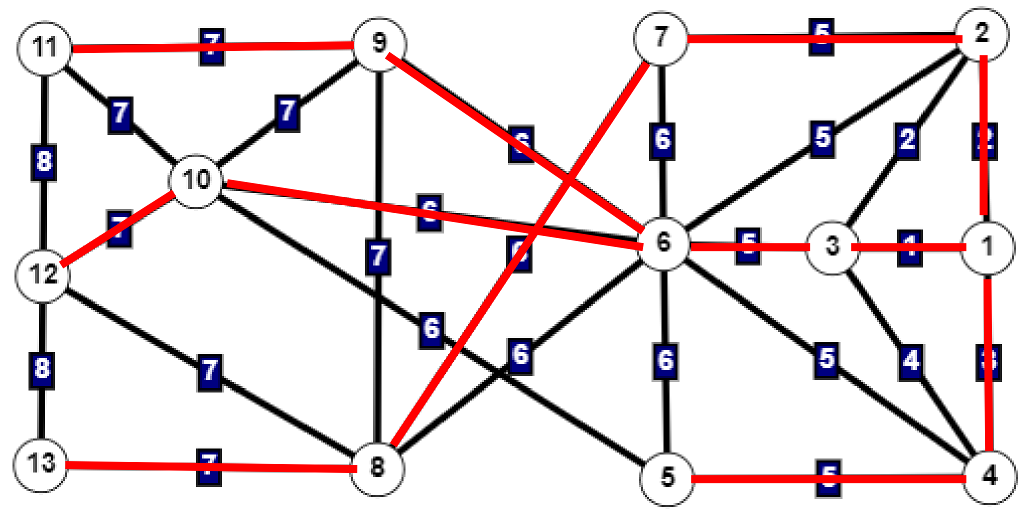
(Phương ND, 2013)

Kiểm nghiệm thuật toán Kruskal

Áp dụng thuật toán Kruskal tìm cây khung nhỏ nhất cho đồ thị được biểu diễn bằng ma trận trọng số như hình dưới?

$$
\left|\begin{matrix}
\infty & 2 & 1 & 3 & \infty & \infty & \infty & \infty & \infty & \infty & \infty & \infty & \infty \\
2 & \infty & 2 & \infty & \infty & 5 & 5 & \infty & \infty & \infty & \infty & \infty & \infty \\
1 & 2 & \infty & 4 & \infty & 5 & \infty & \infty & \infty & \infty & \infty & \infty & \infty \\
3 & \infty & 4 & \infty & 5 & 5 & \infty & \infty & \infty & \infty & \infty & \infty & \infty \\
\infty & \infty & \infty & 5 & \infty & 6 & \infty & \infty & \infty & 6 & \infty & \infty & \infty \\
\infty & 5 & 5 & 5 & 6 & \infty & 6 & 6 & 6 & 6 & \infty & \infty & \infty \\
\infty & 5 & \infty & \infty & \infty & 6 & \infty & 6 & \infty & \infty & \infty & \infty & \infty \\
\infty & \infty & \infty & \infty & \infty & 6 & 6 & \infty & 7 & \infty & \infty & 7 & 7 \\
\infty & \infty & \infty & \infty & \infty & 6 & \infty & 7 & \infty & 7 & 7 & \infty & \infty \\
\infty & \infty & \infty & \infty & 6 & 6 & \infty & \infty & 7 & \infty & 7 & 7 & \infty \\
\infty & \infty & \infty & \infty & \infty & \infty & \infty & \infty & 7 & 7 & \infty & 8 & \infty \\
\infty & \infty & \infty & \infty & \infty & \infty & \infty & 7 & \infty & 7 & 8 & \infty & 8 \\
\infty & \infty & \infty & \infty & \infty & \infty & \infty & 7 & \infty & \infty & \infty & 8 & \infty
\end{matrix}\right|
$$

(Phương ND, 2013)



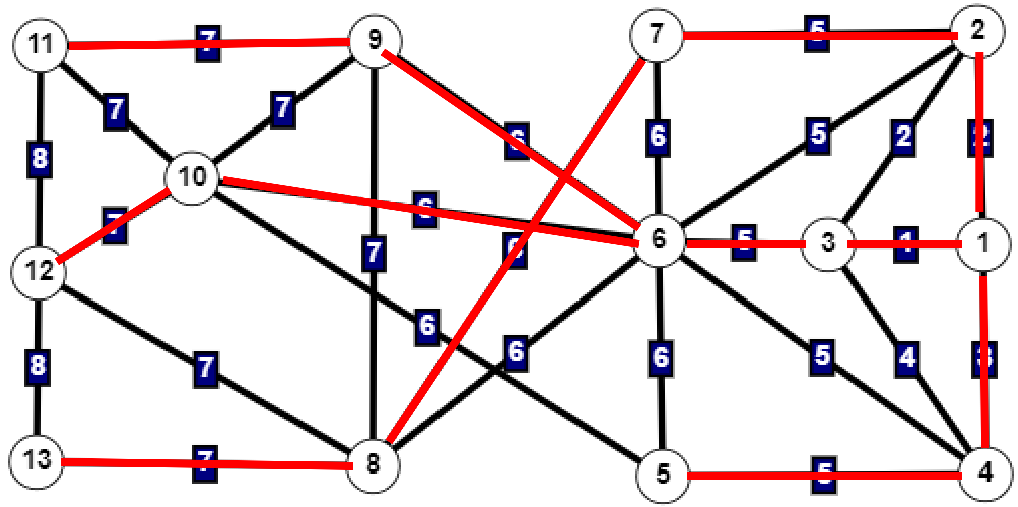
Các cạnh cây khung nhỏ nhất;

Độ dài: 60

1 — 3	6 — 10
1 — 2	7 — 8
1 — 4	6 — 9
3 — 6	10 — 12
4 — 5	9 — 11
2 — 7	8 — 13

Kiểm nghiệm thuật toán Kruskal

```
Do dai cay khung nho nhat: 60
Cac canh cua cay khung nho nhat:
    1    3
    1    2
    1    4
    3    6
    4    5
    2    7
    6   10
    7    8
    6    9
   10   12
    9   11
    8   13
```



Thuật toán Prim (1/2)

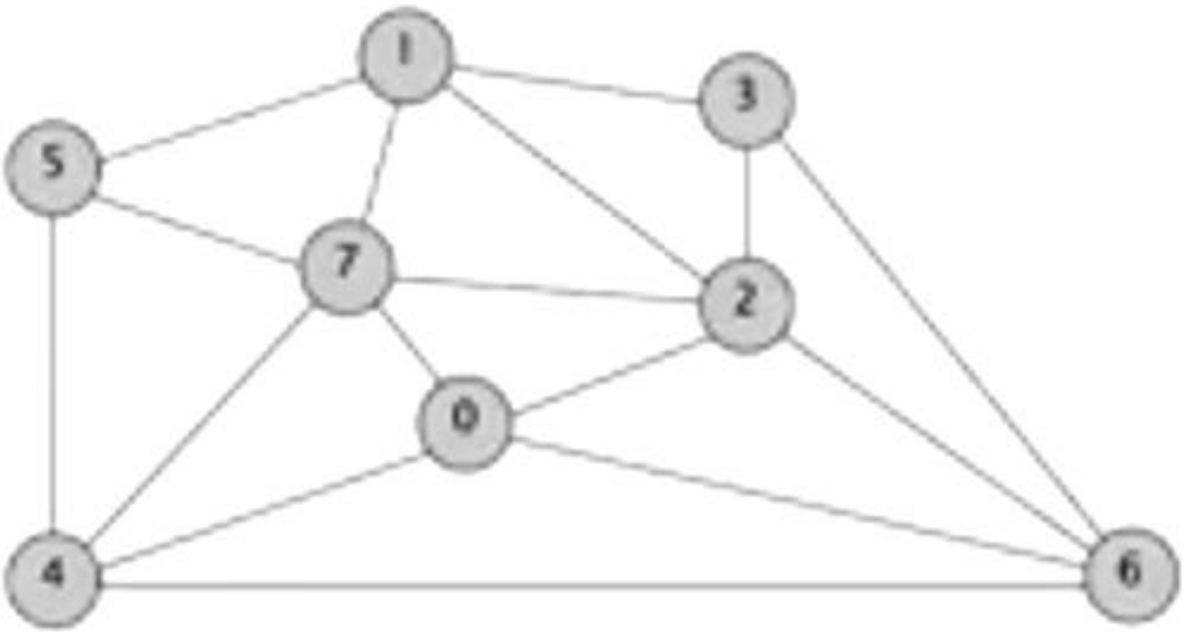
Robert Clay PRIM
(Born 1921)

1. Duy trì 2 tập đỉnh:
   V_H: tập đỉnh của cây khung,
   V: tập các đỉnh $\notin$ cây khung.
   - Ban đầu $V_H = \{s\}$, s là một đỉnh bất kỳ của đồ thị.
   - V = tập đỉnh của đồ thị trừ đi s.
2. Mỗi bước chọn cạnh có trọng số nhỏ nhất và có 1 đỉnh trong V_H và 1 đỉnh trong V.
   - Đưa cạnh này vào cây khung.
   - Đưa đỉnh liền kề với cạnh này từ V sang V_H.
3. Thuật toán dừng lại khi:
   - Cây khung có đủ $(n - 1)$ cạnh,
   - Hoặc không còn đỉnh nào trong V hay $V = \phi$.

Thuật toán Prim (2/2)

```
Prim(s){
    Bước 1 (khởi tạo):
    V_H = {s};                 // Ban đầu V_H chỉ chứa s
    V = V \ {s};               // Loại s ra khỏi V
    T = ∅;                     // Cây khung ban đầu chưa có cạnh nào
    d(H) = 0;                  // Độ dài cây khung ban đầu bằng 0
    Bước 2 (lặp):
    while(V ≠ ∅){
        e = (u, v);            // Cạnh độ dài nhỏ nhất với u ∈ V, v ∈ V_H
        if(không tìm được e)
            return <Đồ thị không liên thông>;
        T = T ∪ {e};           // Đưa e vào cây khung
        d(H) = d(H) + d(e);    // Cập nhật độ dài cây khung
        V_H = V_H ∪ {u};       // Đưa u vào V_H
        V = V \ u;             // Loại u ra khỏi V
    }
    Bước 3 (trả lại kết quả):
    return (T, d(H));
}
```

Minh họa thuật toán Prim

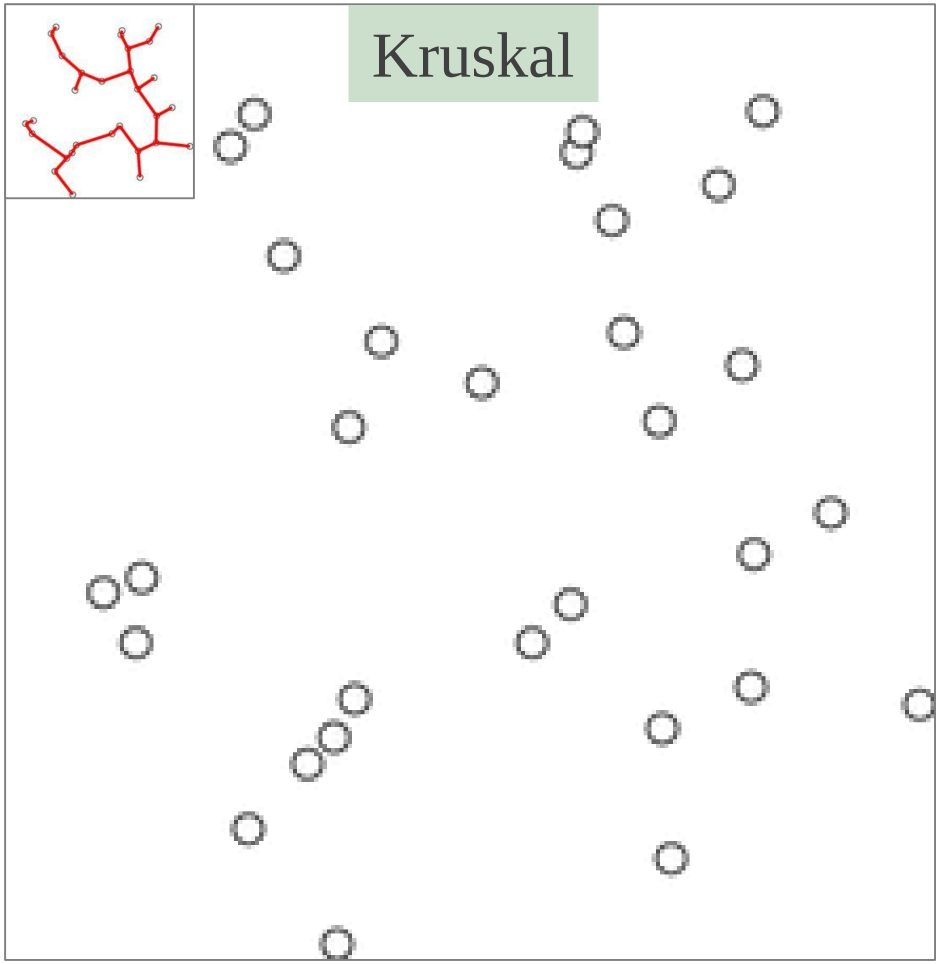


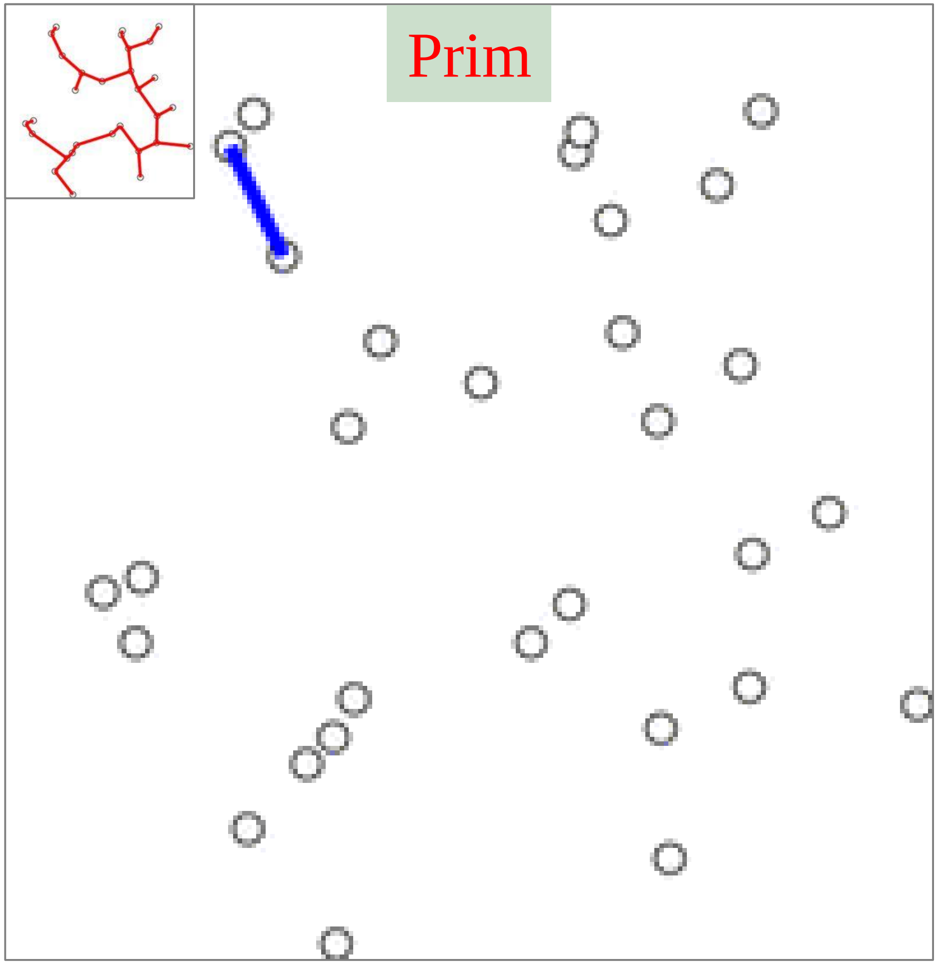
an edge-weighted graph

0-7	0.16
2-3	0.17
1-7	0.19
0-2	0.26
5-7	0.28
1-3	0.29
1-5	0.32
2-7	0.34
4-5	0.35
1-2	0.36
4-7	0.37
0-4	0.38
6-2	0.40
3-6	0.52
6-0	0.58
6-4	0.93

Quan sát thuật toán Prim

Cây bao trùm nhỏ nhất với trọng số là khoảng cách giữa 2 điểm

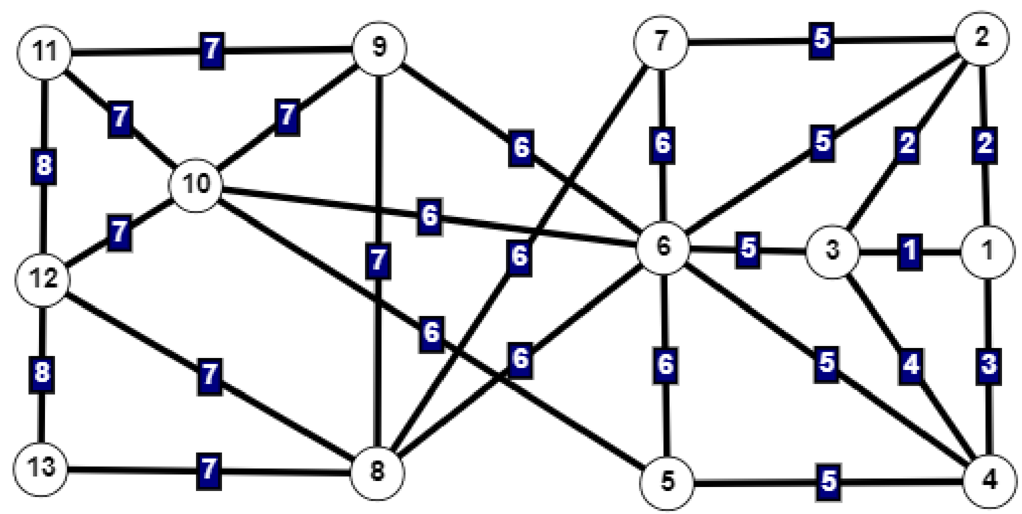




Kiểm nghiệm thuật toán Prim

Áp dụng thuật toán Prim tìm cây khung nhỏ nhất cho đồ thị biểu diễn bằng ma trận trọng số như hình dưới bắt đầu từ đỉnh số 1?

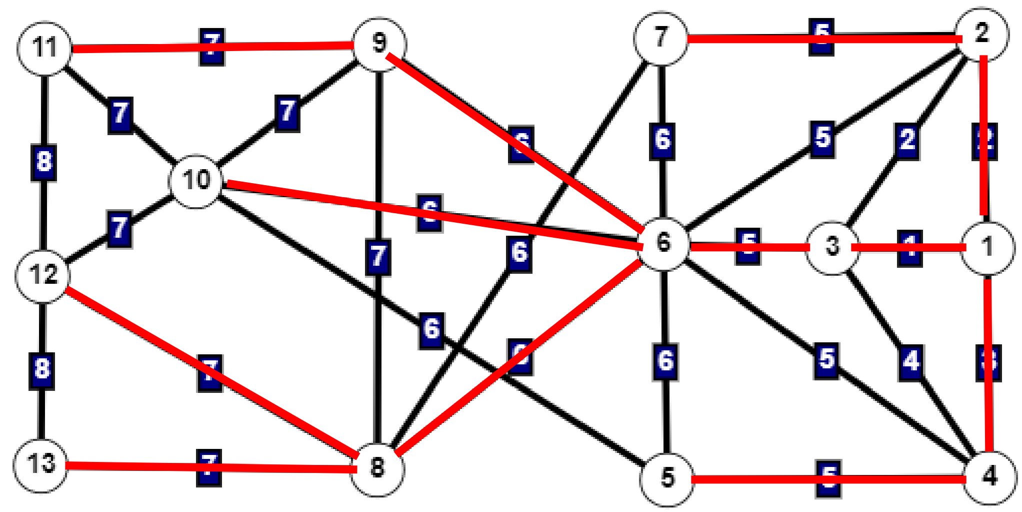
∞	2	1	3	∞	∞	∞	∞	∞	∞	∞	∞	∞
2	∞	2	∞	∞	5	5	∞	∞	∞	∞	∞	∞
1	2	∞	4	∞	5	∞	∞	∞	∞	∞	∞	∞
3	∞	4	∞	5	5	∞	∞	∞	∞	∞	∞	∞
∞	∞	∞	5	∞	6	∞	∞	∞	6	∞	∞	∞
∞	5	5	5	6	∞	6	6	6	6	∞	∞	∞
∞	5	∞	∞	∞	6	∞	6	∞	∞	∞	∞	∞
∞	∞	∞	∞	∞	6	6	∞	7	∞	∞	7	7
∞	∞	∞	∞	∞	6	∞	7	∞	7	7	∞	∞
∞	∞	∞	∞	6	6	∞	∞	7	∞	7	7	∞
∞	∞	∞	∞	∞	∞	∞	∞	7	7	∞	8	∞
∞	∞	∞	∞	∞	∞	∞	7	∞	7	8	∞	8
∞	∞	∞	∞	∞	∞	∞	7	∞	∞	∞	8	∞



(Phương ND, 2013)

Kiểm nghiệm thuật toán Prim

Áp dụng thuật toán Prim tìm cây khung nhỏ nhất cho đồ thị biểu diễn bằng ma trận trọng số như hình dưới bắt đầu từ đỉnh số 1?



∞	2	1	3	∞	∞	∞	∞	∞	∞	∞	∞	∞
2	∞	2	∞	∞	5	5	∞	∞	∞	∞	∞	∞
1	2	∞	4	∞	5	∞	∞	∞	∞	∞	∞	∞
3	∞	4	∞	5	5	∞	∞	∞	∞	∞	∞	∞
∞	∞	∞	5	∞	6	∞	∞	∞	6	∞	∞	∞
∞	5	5	5	6	∞	6	6	6	6	∞	∞	∞
∞	5	∞	∞	∞	6	∞	6	∞	∞	∞	∞	∞
∞	∞	∞	∞	∞	6	6	∞	7	∞	∞	7	7
∞	∞	∞	∞	∞	6	∞	7	∞	7	7	∞	∞
∞	∞	∞	∞	6	6	∞	∞	7	∞	7	7	∞
∞	∞	∞	∞	∞	∞	∞	∞	7	7	∞	8	∞
∞	∞	∞	∞	∞	∞	∞	7	∞	7	8	∞	8
∞	∞	∞	∞	∞	∞	∞	7	∞	∞	∞	8	∞

(Phương ND, 2013)

Các cạnh cây khung nhỏ nhất;

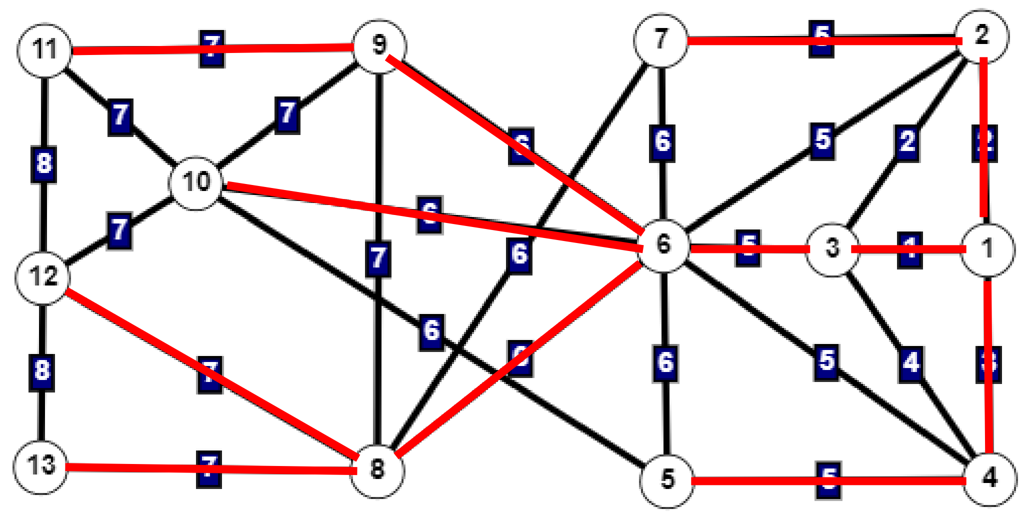
Độ dài: 60

1—3
1—2
1—4
3—6
2—7
4—5

6—8
6—9
6—10
8—12
8—13
9—11

Kiểm nghiệm thuật toán Prim

```
Do dai cay khung nho nhat: 60
Cac canh cua cay khung nho nhat:
   1  3
   1  2
   1  4
   3  6
   2  7
   4  5
   6  8
   6  9
   6 10
   8 12
   8 13
   9 11
```



Tóm tắt

1. Khái niệm cây, các tính chất của cây
2. Cây khung của đồ thị
   - Mọi đồ thị vô hướng liên thông đều có ít nhất một cây khung.
   - Xây dựng cây khung của đồ thị sử dụng các thuật toán BFS và DFS.
3. Bài toán cây khung nhỏ nhất
   - Thuật toán Kruskal và thuật toán Prim

Bài tập

- Cài đặt các thuật toán đã học dựa theo hướng dẫn trong giáo trình;
- Làm các bài tập trong slide bài giảng (download theo link đã được cung cấp);
- Làm các bài tập 1 – 6, Bài tập Chương 5 trong giáo trình.

Kết thúc Bài 5

- Câu hỏi và thảo luận?