Thuật toán DFS và DFS

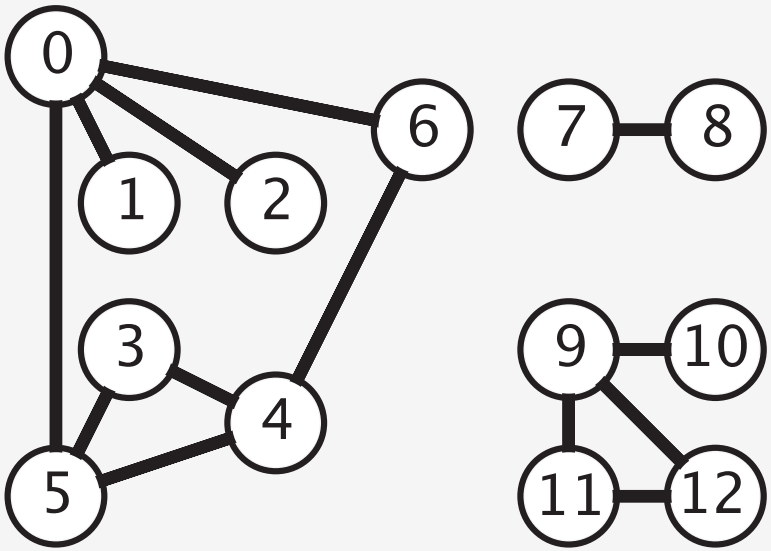
Trần Vĩnh Đức

HUST

Ngày 10 tháng 1 năm 2017

Nội dung

Biểu diễn đồ thị bởi danh sách kề



tinyG.txt

```
13
13
0  5
4  3
0  1
9  12
6  4
5  4
0  2
11  12
9  10
0  6
7  8
9  11
5  3
```

API cho đồ thị

lớp `Graph`	
`Graph(int V)`	*tạo ra một đồ thị rỗng với V đỉnh*
`Graph(istream &in)`	*tạo một đồ thị từ luồng vào in*
`void addEdge(int v, int w)`	*thêm cạnh v-w*
`vector<int> getAdj(int v)`	*danh sách cạnh kề với đỉnh v*
`int getV()`	*số đỉnh*
`int getE()`	*số cạnh*

Ví dụ: Sử dụng lớp Graph

```
//Tinh bac cua dinh v trong G
//G.adj(v).size();
int degree (Graph &G, int v)
{
    int deg = 0;
    for(int w: G.getAdj(v))
        deg++;
    return deg;
}
```

Demo cài đặt lớp Graph.

Nội dung

Tìm kiếm theo chiều sâu (DFS)

Thuật toán DFS cho phép duyệt đồ thị một cách có hệ thống.

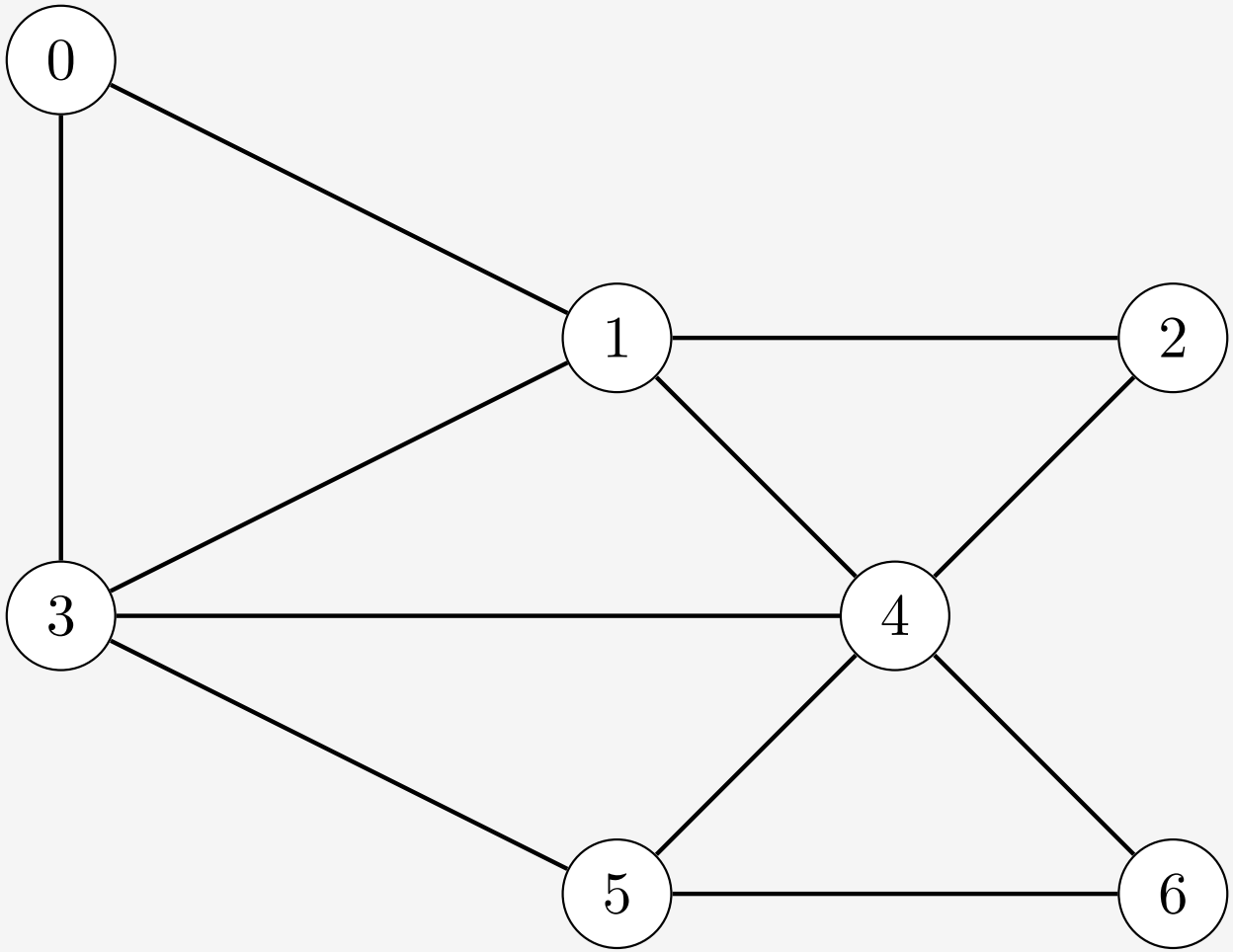
Ứng dụng:

- Kiểm tra xem một đỉnh có liên thông với một đỉnh khác
- Kiểm tra tính liên thông của đồ thị
- Tìm đường đi giữa hai đỉnh

Thuật toán BFS

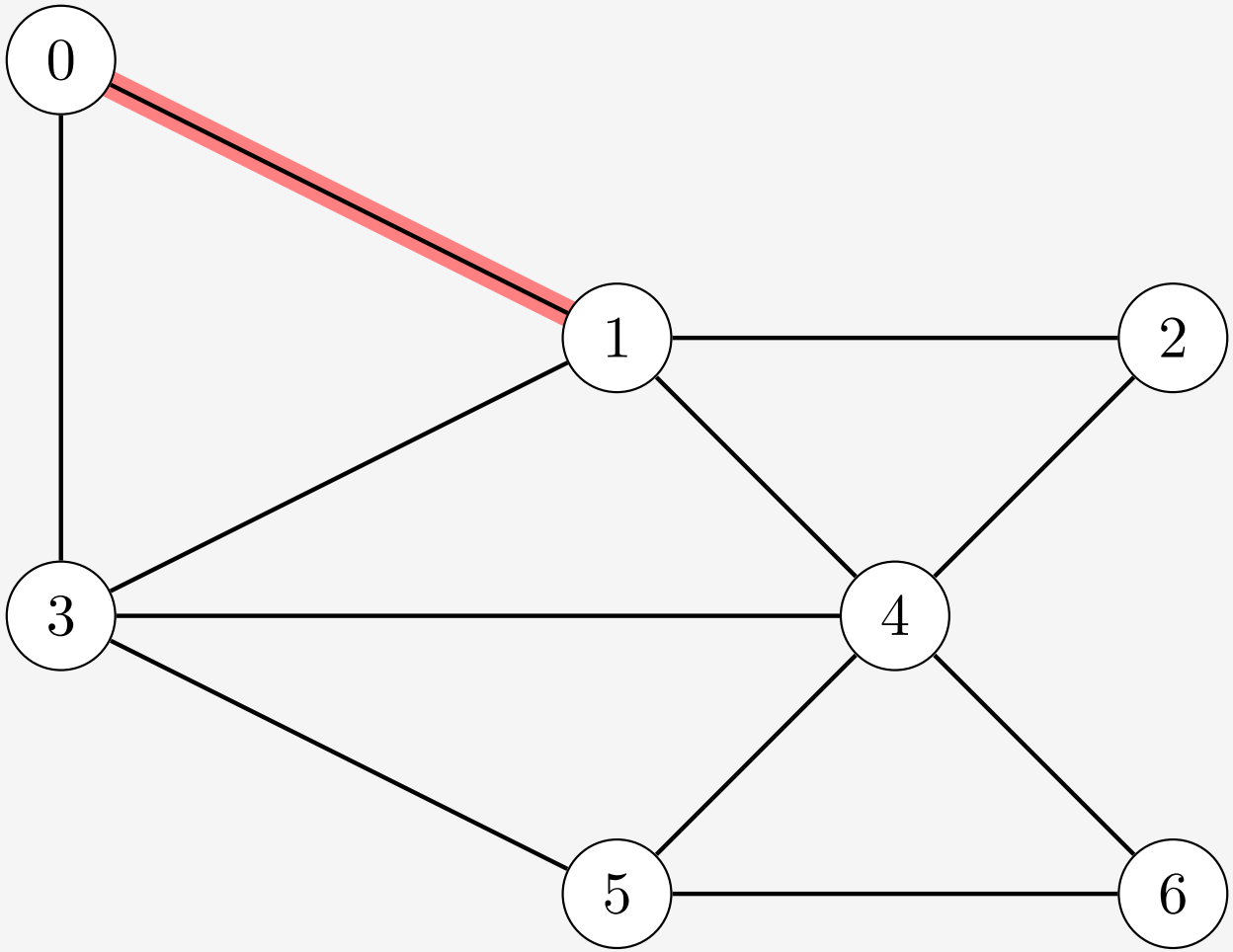
```
DFS (Đồ thị G, đỉnh v):
    Đánh dấu đỉnh v đã được thăm
    Với mỗi đỉnh w  chưa đánh dấu và w kề với v:
        DFS (w)
```

Ví dụ: Thuật toán DFS



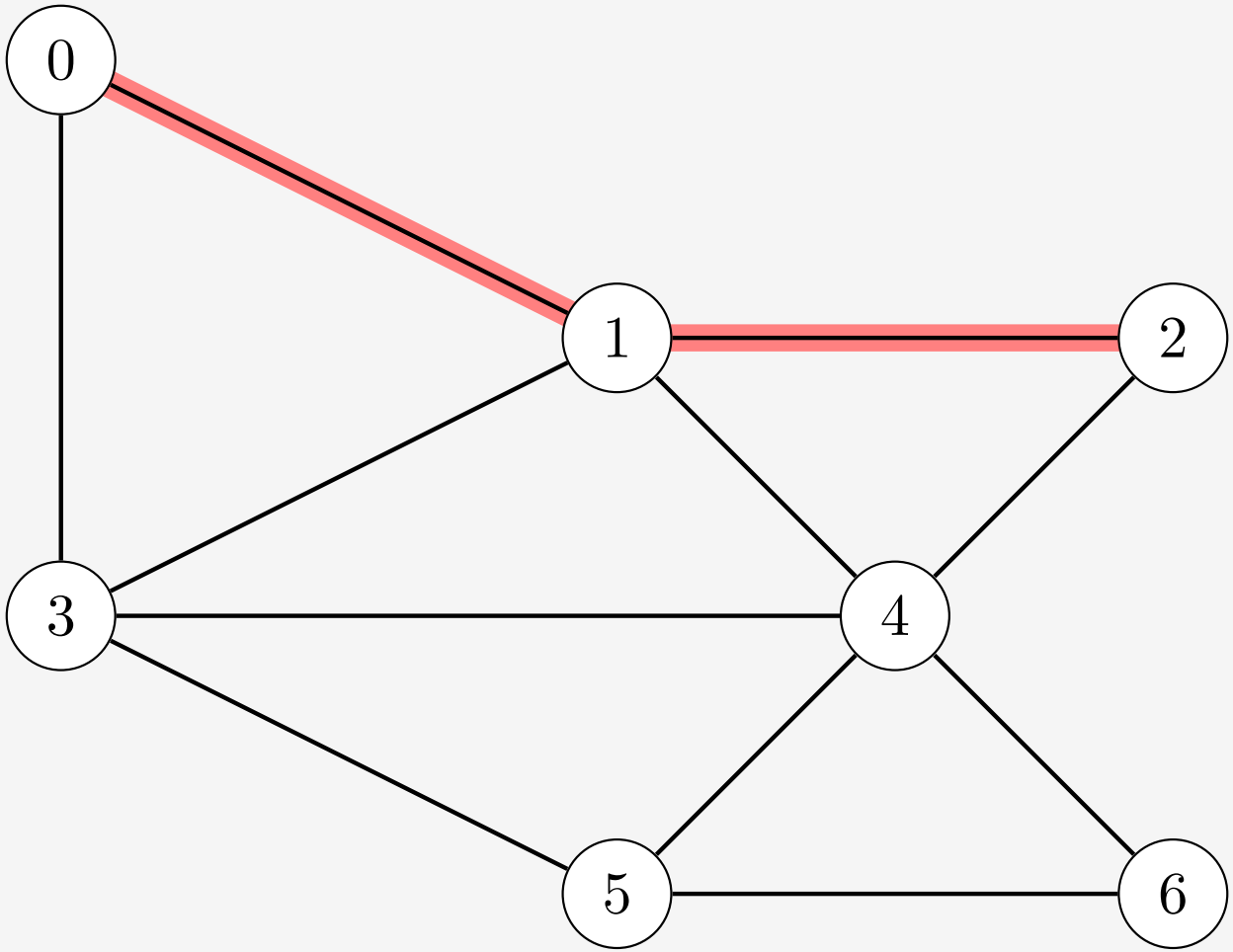
Hình: Thứ tự thăm đỉnh: $0, 1, 2, 4, 3, 5, 6$

Ví dụ: Thuật toán DFS



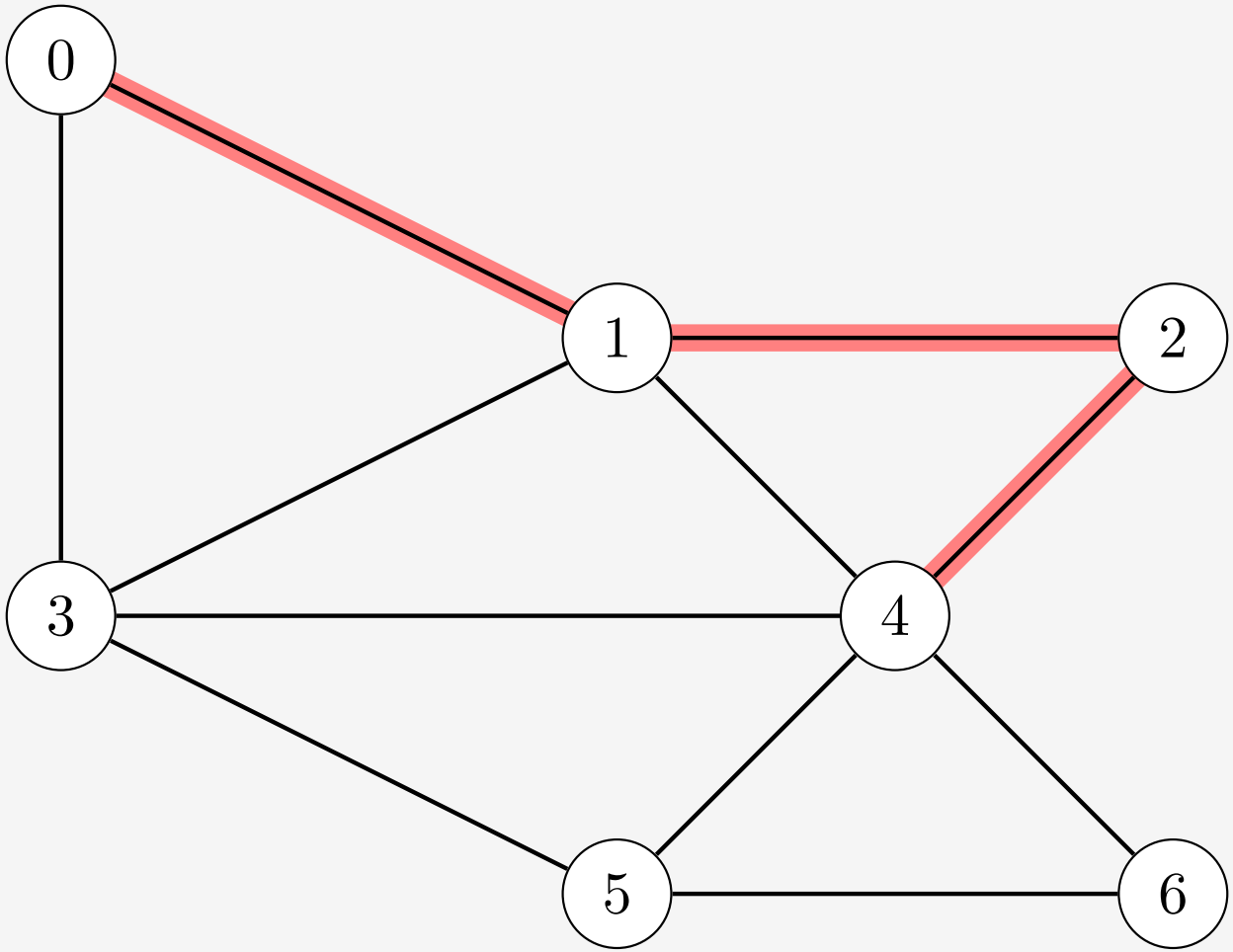
Hình: Thứ tự thăm đỉnh: $0, 1, 2, 4, 3, 5, 6$

Ví dụ: Thuật toán DFS



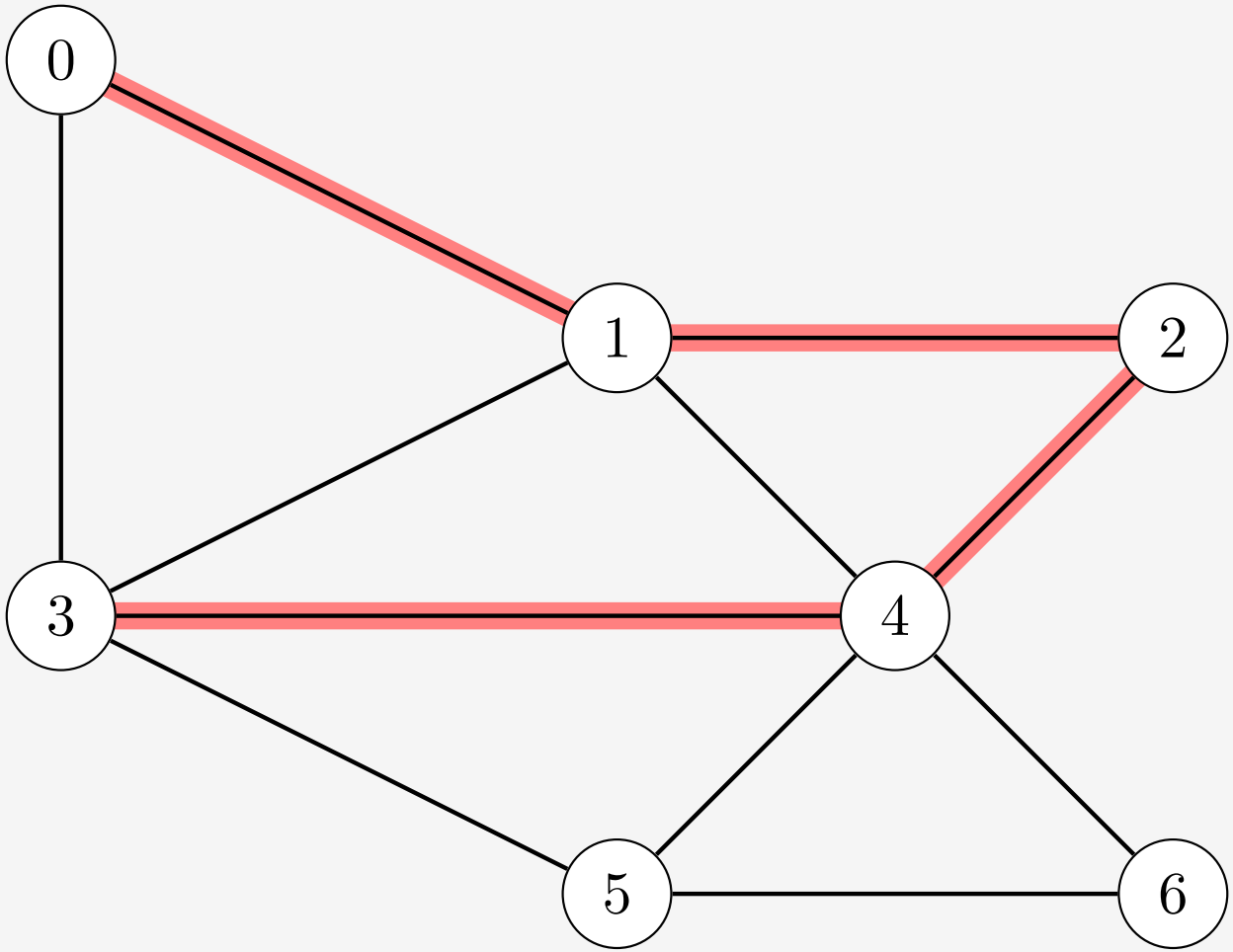
Hình: Thứ tự thăm đỉnh: $0, 1, 2, 4, 3, 5, 6$

Ví dụ: Thuật toán DFS



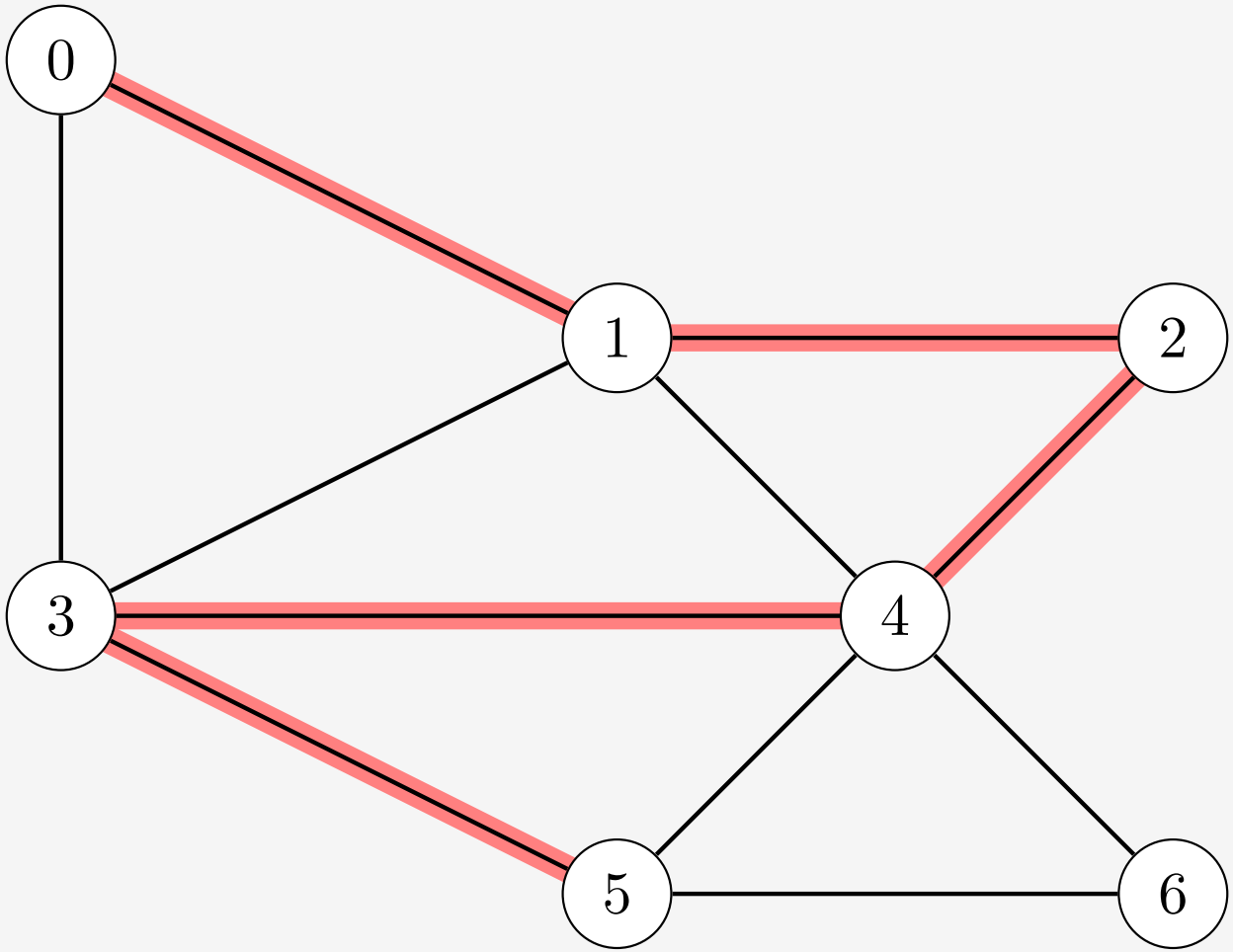
Hình: Thứ tự thăm đỉnh: $0, 1, 2, 4, 3, 5, 6$

Ví dụ: Thuật toán DFS



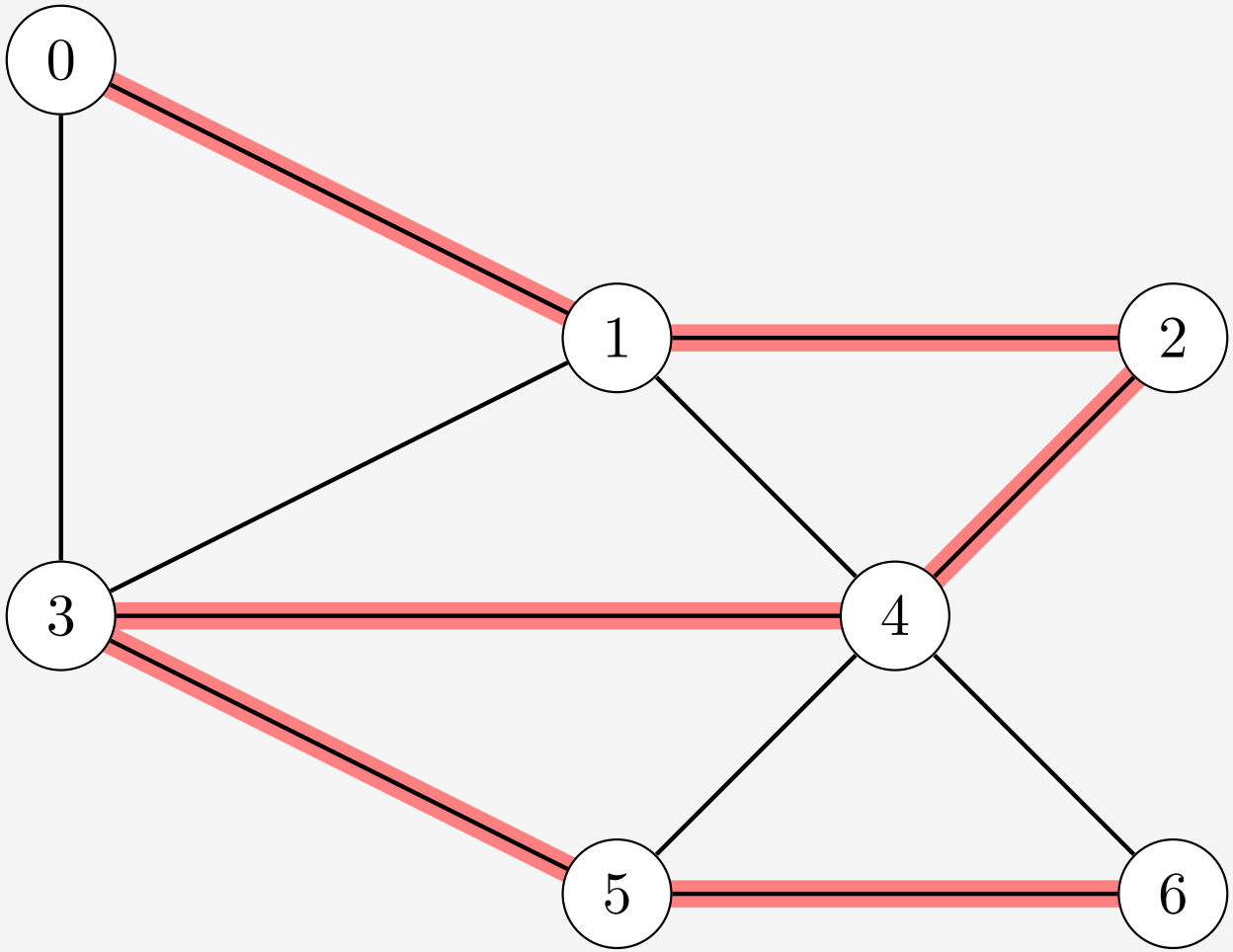
Hình: Thứ tự thăm đỉnh: $0, 1, 2, 4, 3, 5, 6$

Ví dụ: Thuật toán DFS



Hình: Thứ tự thăm đỉnh: $0, 1, 2, 4, 3, 5, 6$

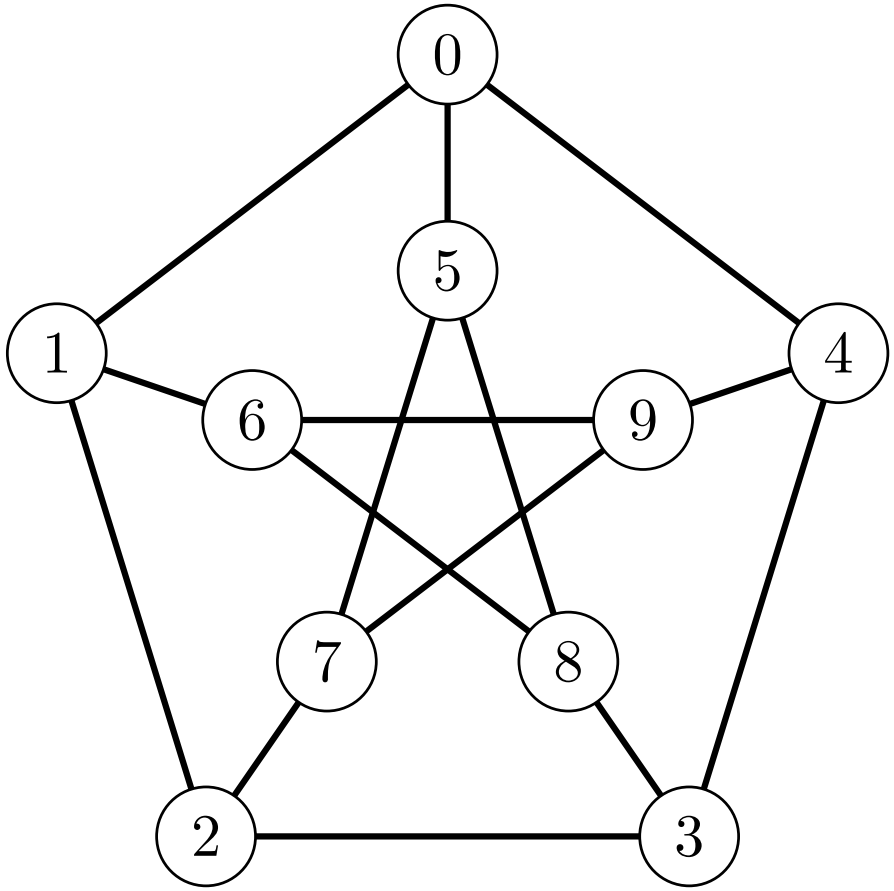
Ví dụ: Thuật toán DFS



Hình: Thứ tự thăm đỉnh: $0, 1, 2, 4, 3, 5, 6$

Bài tập

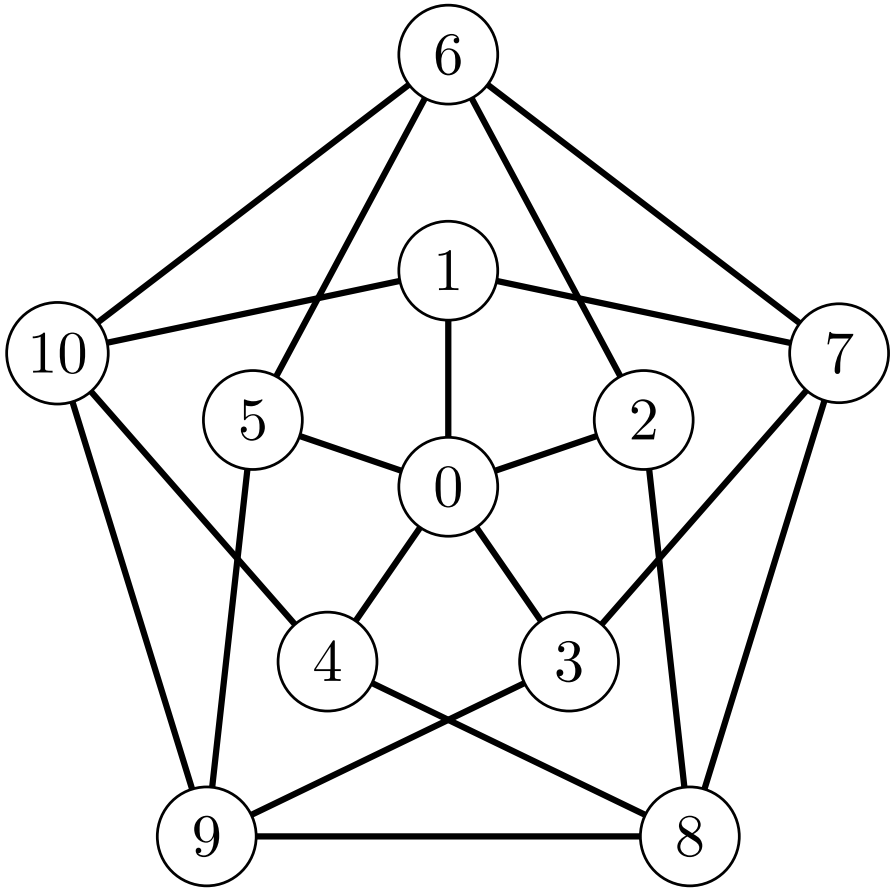
Hãy đưa ra thứ tự thăm đỉnh theo thuật toán DFS đối với đồ thị sau



Hình: Đồ thị Petersen

Bài tập

Hãy đưa ra thứ tự thăm đỉnh theo thuật toán DFS đối với đồ thị sau



Hình: Đồ thị Grötzsch

Cài đặt DFS

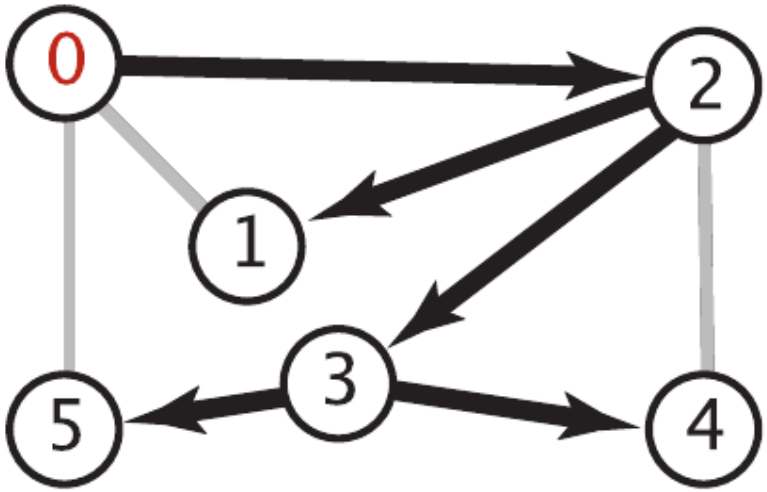
```
void dfs (Graph &G, int v)
{
  marked[v] = true;
  count++;
  for (int w: G.adj(v))
  {
    if(!marked[w]) dfs(G,w);
  }
}
```

- Mảng `bool marked[V]` để đánh dấu các đỉnh đã thăm.
- Biến `int count` dùng để đếm số đỉnh liên thông với đỉnh xuất phát.

Câu hỏi

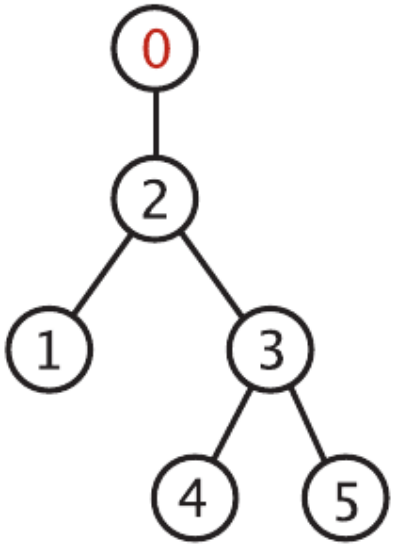
- Làm thế nào để kiểm tra đồ thị có liên thông hay không?
- Làm thế nào để tính số thành phần liên thông của đồ thị?

Tìm đường đi trên đồ thị dùng DFS



edgeTo[]

0	
1	2
2	0
3	2
4	3
5	3



- Sử dụng mảng `edgeTo[V]` thỏa mãn:
  `edgeTo[w] = v` nếu v–w là cạnh tới w lần đầu tiên.
- Đường đi từ 0–5 là : 5 - 3 - 2 - 0

Cài đặt DFS để tìm đường đi 1

```
void dfs (Graph &G, int v)
{
  marked[v] = true;
  for(int w: G.adj(v))
      if (!marked[w])
      {
        //Luu lai canh toi w lan dau v-w
        edgeTo[w] = v;
        dfs (G,w);
      }
}
```

Tìm đường đi tới một đỉnh

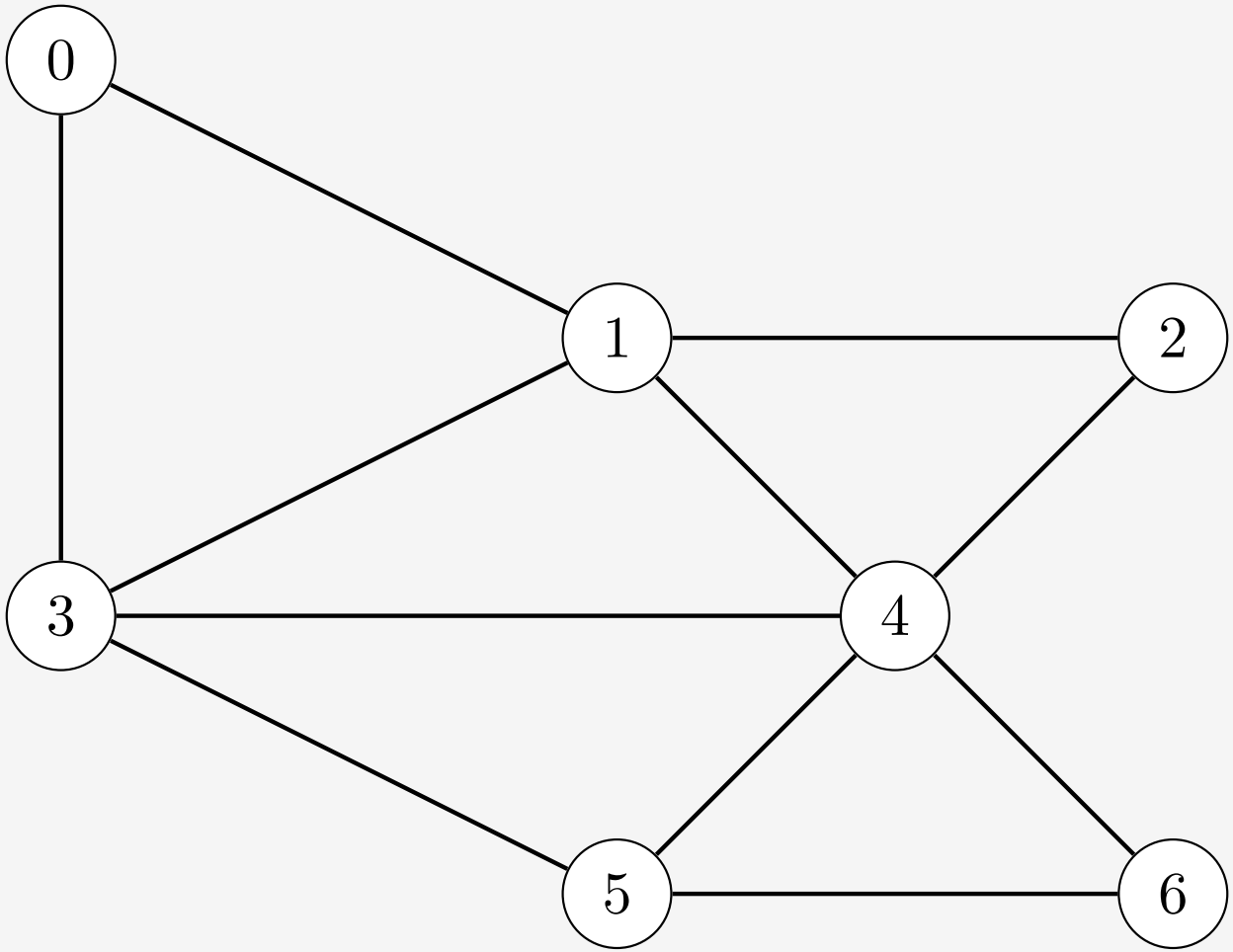
```
// Danh sach cac dinh tren duong di toi v
vector<int> pathTo(int v)
{
  vector <int> path;
  if (!hasPathTo(v))
      return path;

  for (int x = v; x != s; x = edgeTo[x])
      path.push_back(x);

  path.push_back(s);

  return path;
}
```

Nội dung

Thuật toán BFS

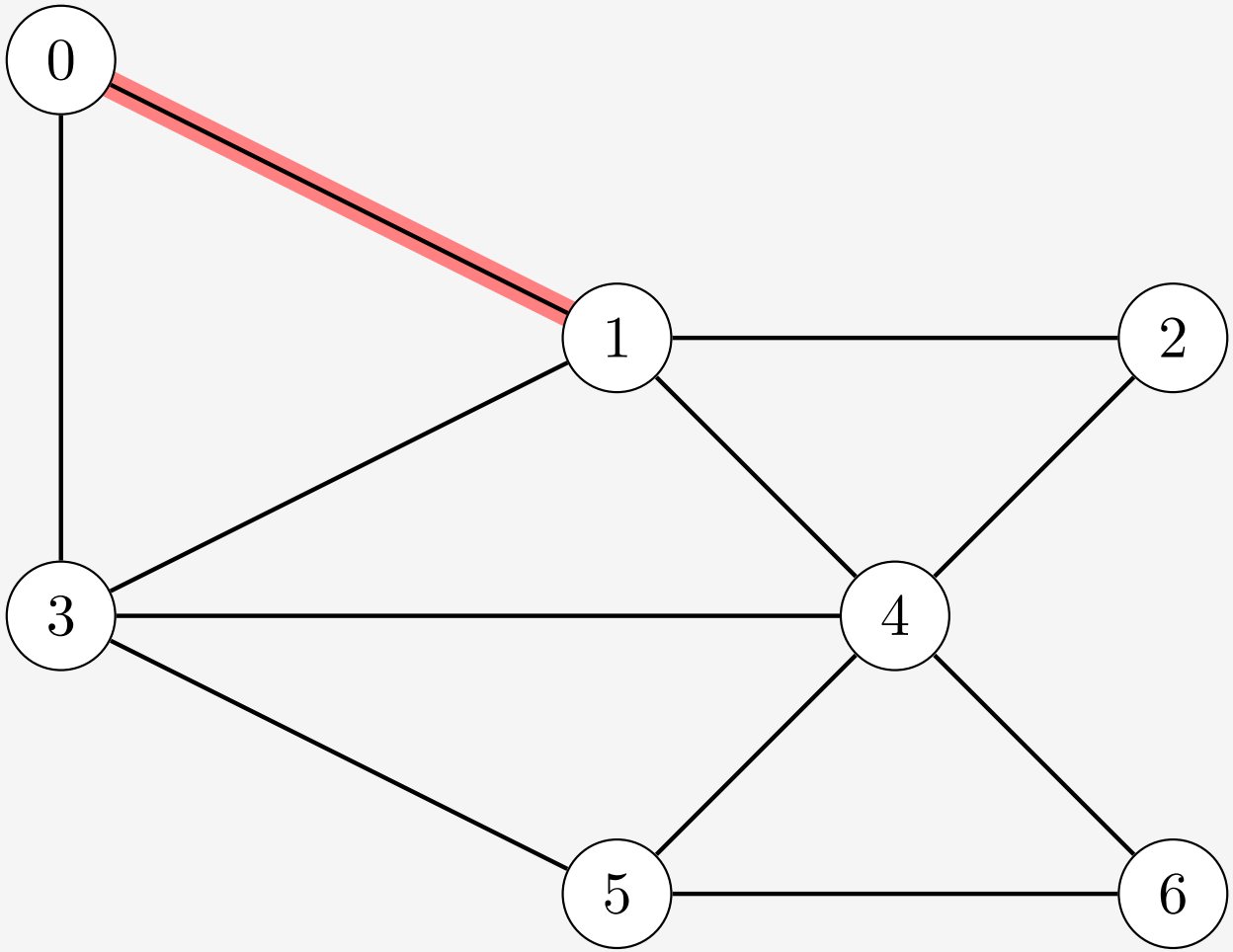
```
BFS (Đồ thị G, đỉnh s):
    Đánh dấu đỉnh s đã được thăm
    Đưa đỉnh s vào hàng đợi
    while (hàng đợi chưa rỗng):
        Xóa đỉnh v tiếp theo trong hàng đợi
        Đưa vào hàng đợi mọi đỉnh chưa được đánh dấu
         và kề với v  và đánh dấu chúng
```

Ví dụ: Thuật toán BFS



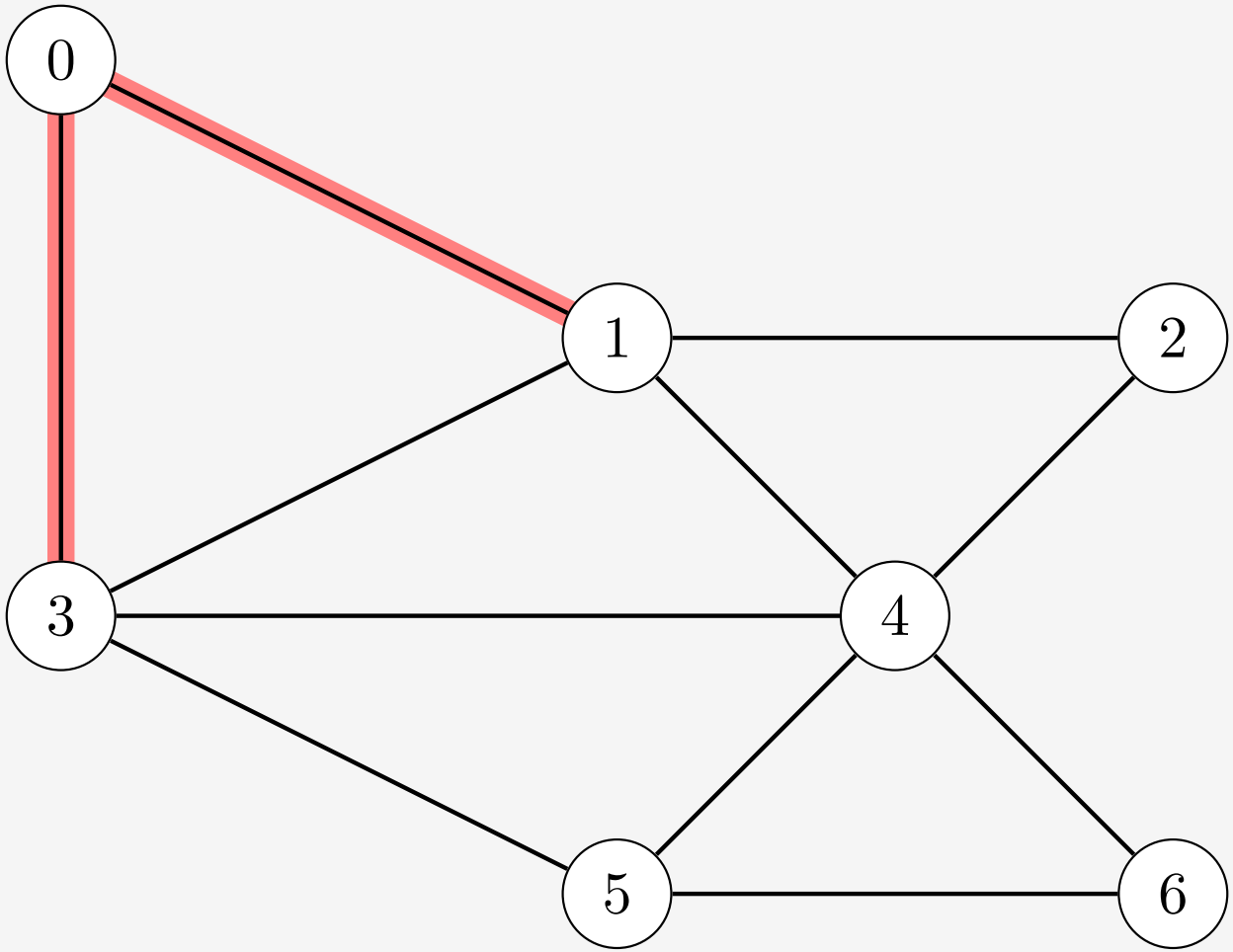
Hình: Thứ tự thăm đỉnh : $0, 1, 3, 2, 4, 5, 6$

Ví dụ: Thuật toán BFS



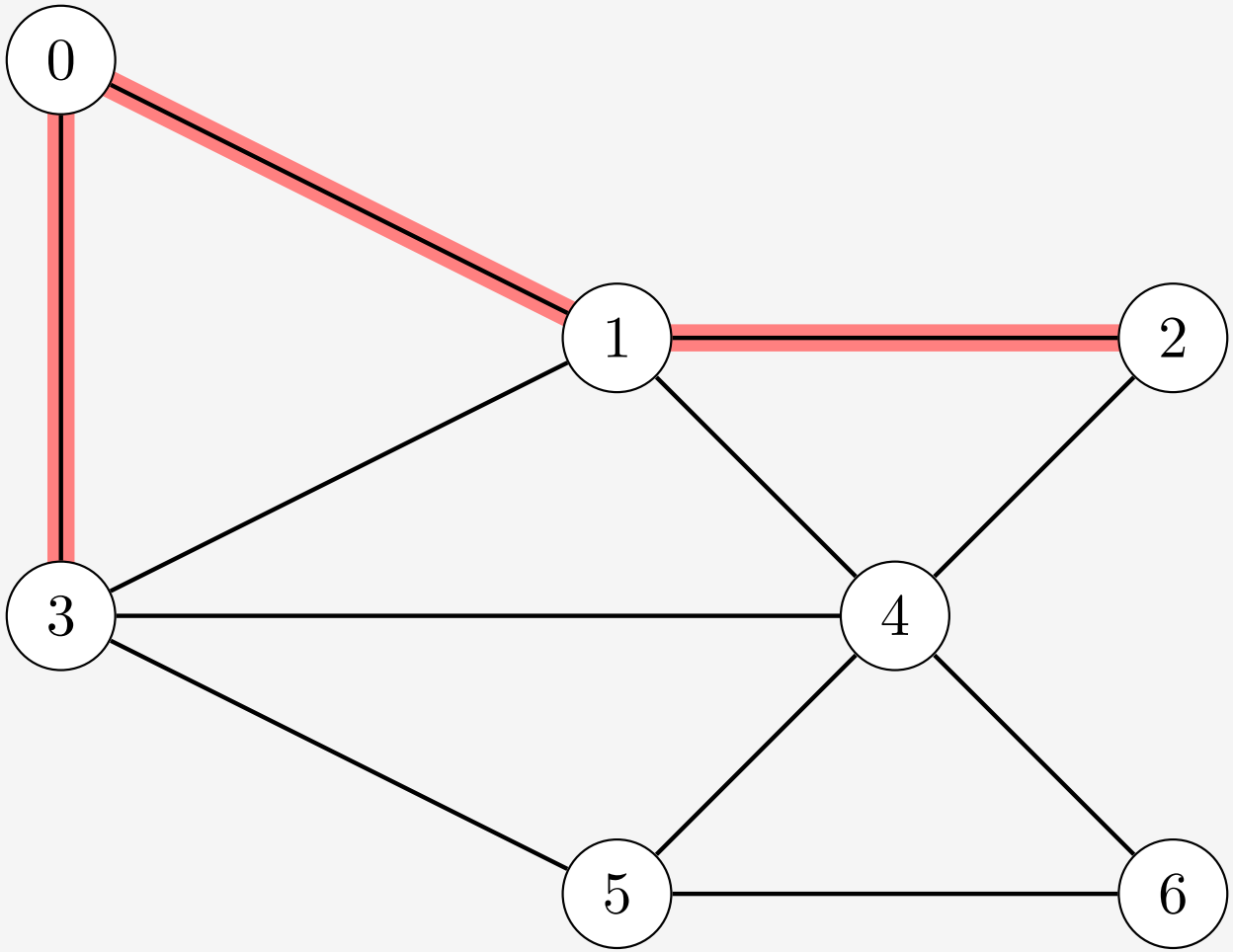
Hình: Thứ tự thăm đỉnh : $0, 1, 3, 2, 4, 5, 6$

Ví dụ: Thuật toán BFS



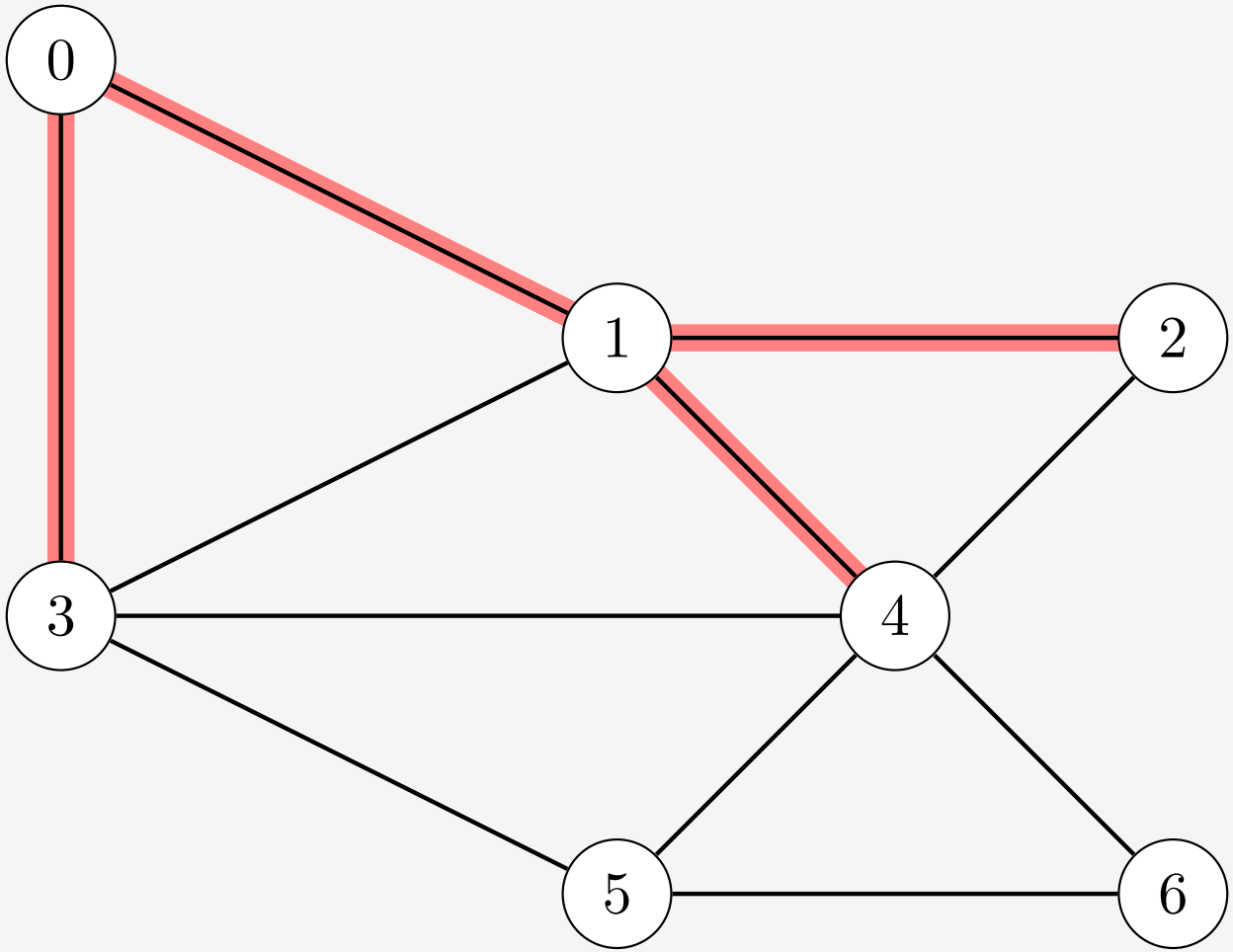
Hình: Thứ tự thăm đỉnh : $0, 1, 3, 2, 4, 5, 6$

Ví dụ: Thuật toán BFS



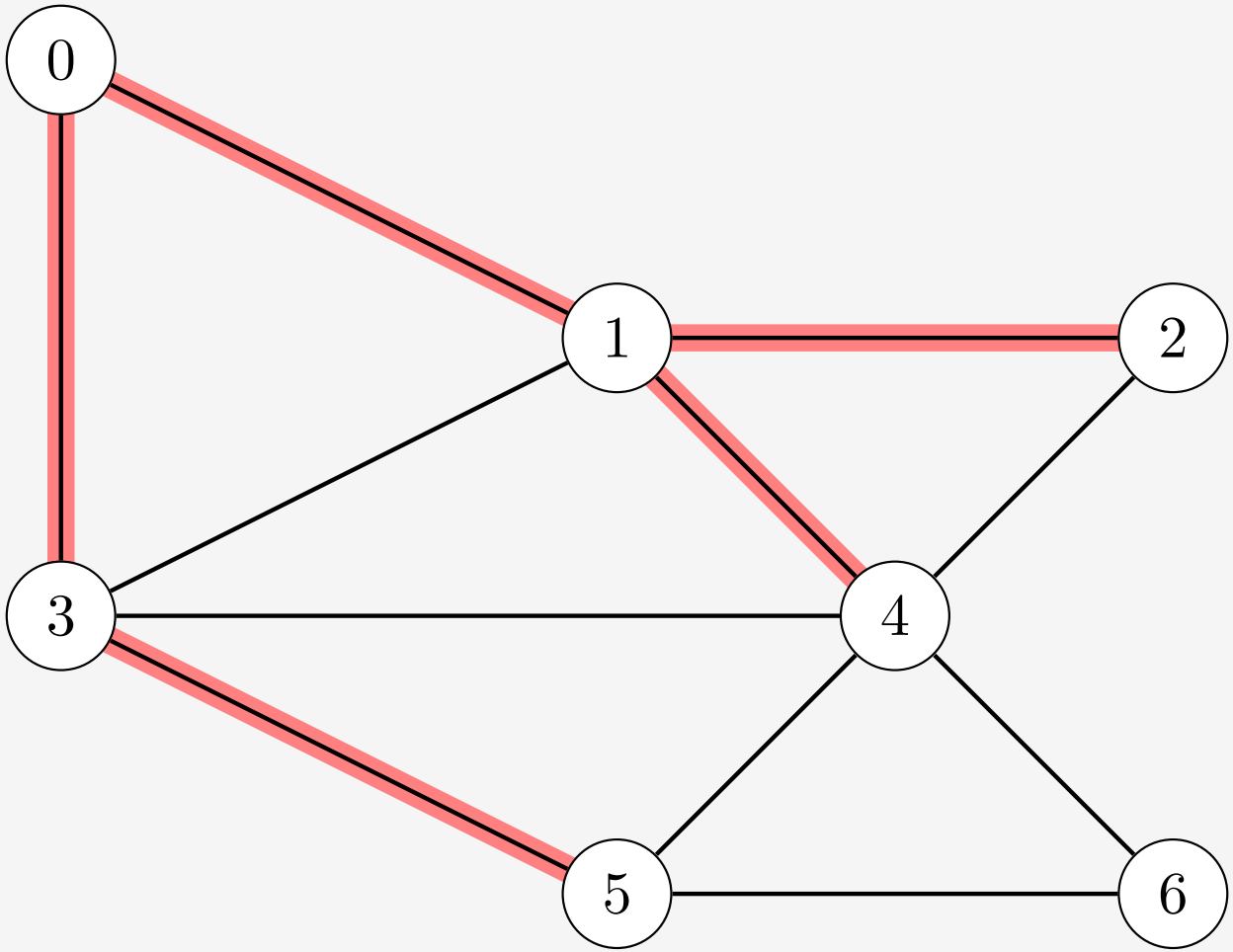
Hình: Thứ tự thăm đỉnh : $0, 1, 3, 2, 4, 5, 6$

Ví dụ: Thuật toán BFS



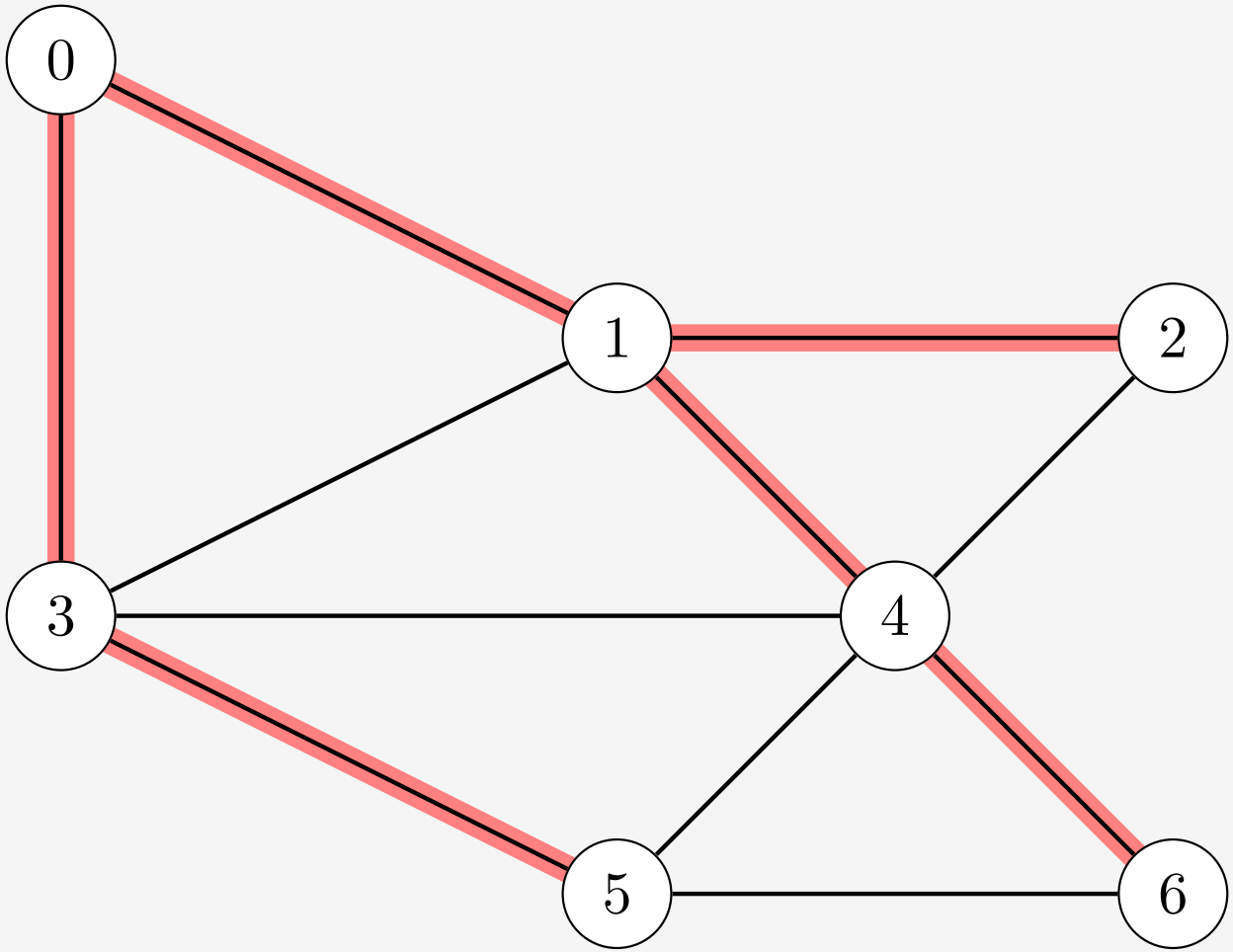
Hình: Thứ tự thăm đỉnh : $0, 1, 3, 2, 4, 5, 6$

Ví dụ: Thuật toán BFS



Hình: Thứ tự thăm đỉnh : $0, 1, 3, 2, 4, 5, 6$

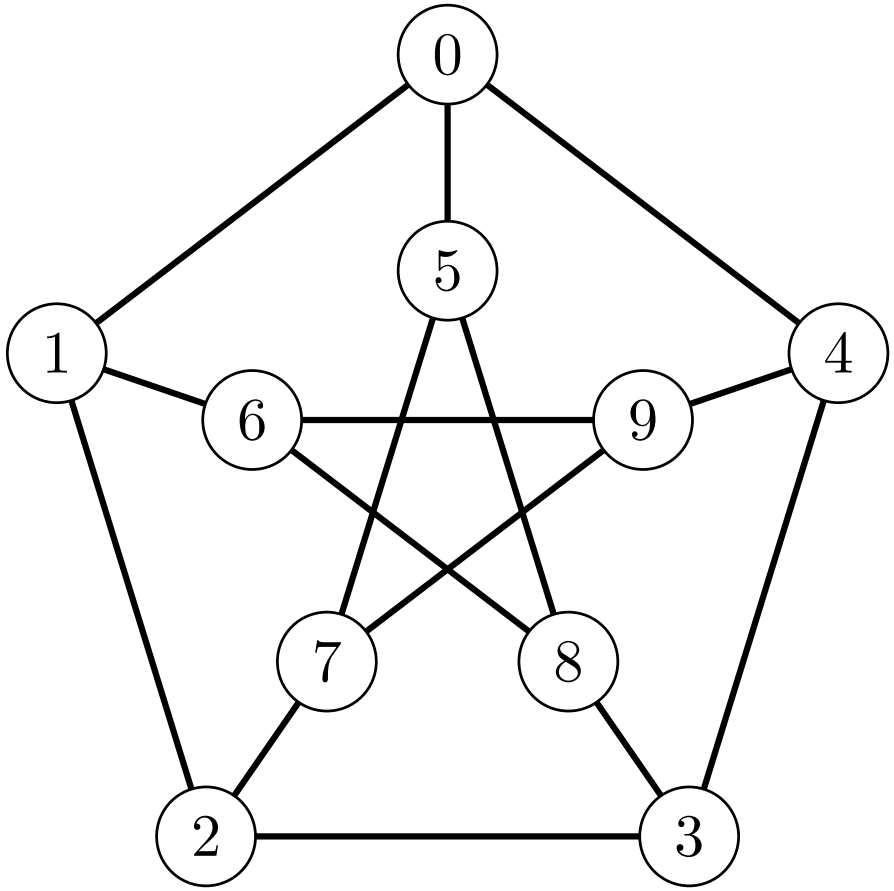
Ví dụ: Thuật toán BFS



Hình: Thứ tự thăm đỉnh : $0, 1, 3, 2, 4, 5, 6$

Bài tập

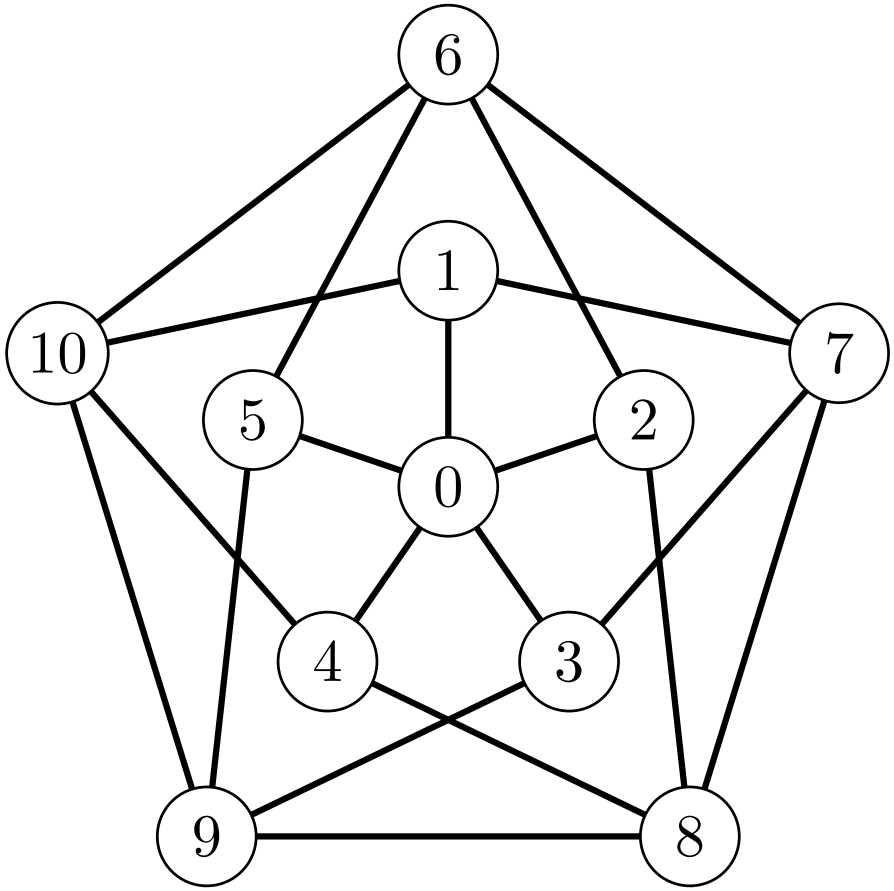
Hãy đưa ra thứ tự thăm đỉnh theo thuật toán BFS đối với đồ thị sau



Hình: Đồ thị Petersen

Bài tập

Hãy đưa ra thứ tự thăm đỉnh theo thuật toán BFS đối với đồ thị sau

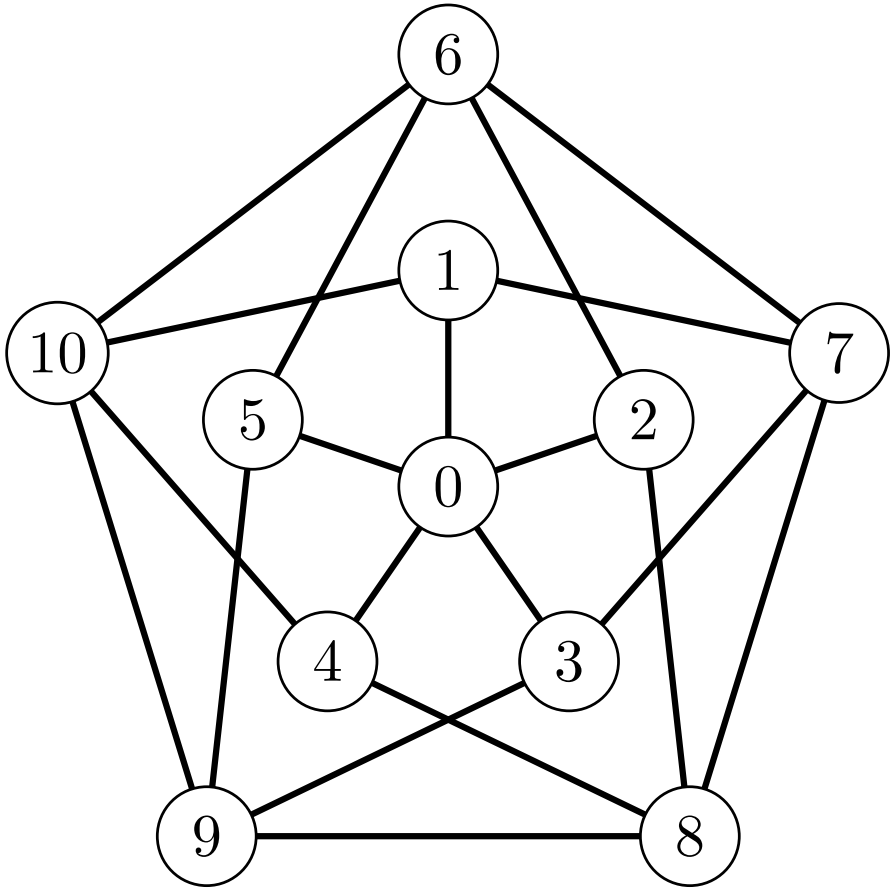


Hình: Đồ thị Grötzsch

```
void bfs (Graph &G, int s)
{
   queue<int> Q;
   marked[s] = true;
   Q.push(s);
   while (!Q.empty())
   {
      int v = Q.front();
      Q.pop();
      for(int x: G.adj(v))
          if (!marked[x])
          {
             edgeTo[x] = v;
             marked[x] = true;
             Q.push(x);
          }
    }
 }
```

Bài tập

Hãy đưa ra mảng edgeTo[] sau chi chạy theo thuật toán BFS với đồ thị sau đây. Từ mảng đó, hãy tìm một đường đi từ đỉnh 0 đến đỉnh 7.



Hình: Đồ thị Grötzsch

Mệnh đề

Với mỗi đỉnh v có thể tới được từ s, thuật toán BFS tính một đường đi ngắn nhất từ s tới v (không có đường đi nào từ s đến v với ít cạnh hơn).