DUYỆT ĐỒ THỊ ƯU TIÊN CHIỀU RỘNG
(Breadth First Search)

I. Các kiểu biểu diễn đồ thị

I.1 Biểu diễn đồ thị bằng danh sách cạnh

```
int m;                          // Số cạnh
pair<int,int> E[maxm];          // Danh sách cạnh
```

I.2 Biểu diễn đồ thị bằng mảng vector

```
int n;                          // Số đỉnh
vector<vector<int> > adj;       // vector kề
```

Code chuyển từ biểu diễn danh sách cạnh sang biểu diễn bằng mảng vector:

```
adj.resize(n+1);
for(int i=1;i<=m;++i) {
    int u=E[i].first, v=E[i].second;
    adj[u].push_back(v);
    adj[v].push_back(u);   // Nếu đồ thị là vô hướng
}
```

II. Hàm duyệt chiều rộng (BFS):

```
void BFS(int xp, vector<int> &d, vector<int> p) {
    queue<int> q;
    d.resize(n+1,INF);
    d[s]=0; p[s]=0;
    q.push(s);
    while (!q.empty()) {
        int u=q.front();
        q.pop();
        for(int v : adj[u]) {
            if (d[v]==INF) {
                d[v]=d[u]+1; p[v]=u;
                q.push(v);
            }
       }
    }
}
```

Chú ý thủ tục BFS trả về:

- Mảng d[1...n] - mảng khoảng cách (số cạnh) từ s đến các đỉnh khác (bằng INF nếu không đi đến được)
- Mảng p[1...n] - mảng tạo nên cây BFS. Mảng này cũng dùng để tìm 1 đường đi ngắn nhất

III. Các ứng dụng của BFS

1. Tìm một đường đi ngắn nhất từ đỉnh s đến đỉnh t

```
if (d[t]==INF) {
    cout << "No path!";
} else {
    vector<int> path;
```

```
    for(int v=t; v!=0; v=p[v]) path.push_back(v);
    reverse(path.begin(), path.end());
    cout << "Path: ";
    for(int v : path) cout << v << ' ';
}
```

2. Tìm các thành phần liên thông

```
int LT[maxn];

void BFS(int xp,int id) {
    queue<int> q;
    LT[s]=id;
    q.push(s);
    while (!q.empty()) {
        int u=q.front();
        q.pop();
        for(int v : adj[u]) {
            if (LT[v]==0) {
                LT[v]=id;
                q.push(v);
            }
       }
    }
}

int main() {
    ...
    for(int u=1;u<=n;++u) LT[u]=0;
    id=0;
    for(int u=1;u<=n;++u) if (LT[u]==0) BFS(u,++id);
    ...
}
```

3. Đếm số lượng đường đi ngắn nhất từ s

```
void BFS(int xp, vector<int> &d, vector<int> &cnt) {
    queue<int> q;
    d.resize(n+1,INF);
    cnt.resize(n+1,0);
    d[s]=0; cnt[s]=1;
    q.push(s);
    while (!q.empty()) {
        int u=q.front();
        q.pop();
        for(int v : adj[u]) {
            if (d[v]==INF) {
                d[v]=d[u]+1; cnt[v] = cnt[u];
                q.push(v);
            } else if (d[v]==d[u]+1)
                cnt[v] = cnt[v] + cnt[u];
       }
    }
}
```

Mảng cnt[1...n] cho biết số lượng đường đi ngắn nhất từ đỉnh s đến đỉnh u.