

**ĐẠI HỌC QUỐC GIA TP.HCM**  
**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**



CS112.N21.KHTN

**BÀI TẬP TUẦN 4**  
**THUẬT TOÁN BRUTE FORCE**

**Nhóm : 14**  
**GV hướng dẫn : Nguyễn Thanh Sơn**

Tp.HCM ngày 13 tháng 4 năm 2023

# 1 Bài tập: Bipartite Graph

A graph is said to be bipartite if all its vertices can be partitioned into two disjoint subsets  $X$  and  $Y$  so that every edge connects a vertex in  $X$  with vertex in  $Y$ . (One can also say that a graph is bipartite if its vertices can be colored in two colors so that every edge has its vertices colored in different colors; such graphs are also called 2-colorable)

1. Design a DFS-based algorithm for checking whether a graph is bipartite
2. Design a BFS-based algorithm for checking whether a graph is bipartite

## Trả lời

### 1.1 Design a DFS-based algorithm for checking whether a graph is bipartite

Thuật toán:

- Dùng 1 mảng colors để lưu trữ 2 giá trị 0 hoặc 1 để biểu diễn 2 giá trị màu.
- Gọi DFS để duyệt qua tất cả các node.
- Nếu tại node  $u$  mà DFS chưa duyệt qua lần nào, thì ta sẽ gán giá trị màu cho node  $u$  này là đối của node  $v$  ( $!color[v]$ ) và gọi DFS để duyệt tới node liên kết với node  $u$ .
- Nếu tại bất cứ điểm nào, mà giá trị màu của node  $u$  bằng với giá trị màu của node  $v$  thì node đó không bipartite dẫn đến đồ thị đó không phải là Bipartite Graph.

---

```
bool isBipartite(vector<int> adj[], int v,
                vector<bool>& visited, vector<int>& color)
{
    for (int u : adj[v]) {
        if (visited[u] == false) {
            visited[u] = true;
            color[u] = !color[v];

            if (!isBipartite(adj, u, visited, color))
                return false;
        }

        else if (color[u] == color[v])
            return false;
    }
    return true;
}
```

---

## 1.2 Design a BFS-based algorithm for checking whether a graph is bipartite

Thuật toán:

- Dùng 1 mảng colors để lưu trữ 2 giá trị 0 hoặc 1 để biểu diễn 2 giá trị màu.
- Gán 1 giá trị màu mặc định x cho đỉnh nguồn u.
- Tô màu cho tất cả các đỉnh v có cạnh từ u đến v với giá trị là giá trị đối của giá trị vừa gán. (đặt vào tập V)
- Lại tiếp tục tô màu cho tất cả các đỉnh u có cạnh từ v đến u với giá trị là giá trị đối của giá trị vừa gán. (đặt vào tập U).
- Tiếp tục tô màu cho đến khi tất cả các đỉnh đều được tô.
- Nếu tại bất cứ điểm nào, mà giá trị đỉnh đang tô hiện tại lại bằng giá trị điểm tô trước đó thì kết luận đồ thị đó không phải Bipartite Graph.

---

```
bool isBipartite(int G[][V], int src)
{
    int colorArr[V];
    for (int i = 0; i < V; ++i)
        colorArr[i] = -1;

    colorArr[src] = 1;

    queue <int> q;
    q.push(src);

    while (!q.empty())
    {
        int u = q.front();
        q.pop();

        if (G[u][u] == 1)
            return false;

        for (int v = 0; v < V; ++v)
        {
            if (G[u][v] && colorArr[v] == -1)
            {
                colorArr[v] = 1 - colorArr[u];
                q.push(v);
            }

            else if (G[u][v] && colorArr[v] == colorArr[u])
                return false;
        }
    }

    return true;
}
```

---