**Nhóm 3:**

Mai Nhất Tùng - 21520523

Hà Trọng Tài - 21520436

Nguyễn Ngọc Nhung – 21521248

BÀI LÀM

Text, letter

Description automatically generated

1. Ý tưởng:

* Dùng 1 mảng biểu diễn giá trị màu.
* Gọi DFS để duyệt qua tất cả các node.
* Xét danh sách node u là node có liên kết với node v:
  + Nếu tại node u mà DFS chưa duyệt qua lần nào, thì ta sẽ gán giá trị màu cho node u này là đối 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.

Code:

|  |
| --- |
| bool isBipartite(vector<int> pos[], int v, vector<bool> &check, vector<int> &color){  for(int u : pos[v]) {  if(check[u] == false) {  check[u] = true;  color[u] = !color[v];  if(!isBipartite(pos, u, check, color))  return false;  }  else if(color[u] == color[v])  return false;  }  return true  } |

1. Ý tưởng:

* Dùng 1 mảng biểu diễn giá trị màu
* Gán giá trị màu mặc định x cho đỉnh u:
* Tô màu cho tất cả các đỉnh v có cạnh từ u đến v với giá trị là giá trị đối với x
* Sau đó thực hiện tô màu cho các đỉnh liên kết với v với giá trị màu đối của đỉnh v. Thực hiện lặp đi lặp lại cho đến khi tất cả các đỉnh đều được tô
* Nếu tại đỉnh nào có giá trị màu bằng với giá trị màu của đỉnh tô trước đó thì đồ thị đó không phải Bipartite Graph

Code:

|  |
| --- |
| bool isBipartite(int G[][], int n, int root) {  int color[n];  for(int i = 0; i < n; i++)  color[i] = -1;  color[root] = 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 < n; v++) {  if(G[u][v] && color[v] == -1) {  color[v] = !color[u];  q.push(v);  }  else if(G[u][v] && color[v] == color[u])  return false;  }  }  return true;  } |