

Báo cáo chương trình tô màu đỉnh

Ngày 17 tháng 6 năm 2025

Mục lục

1	Giới thiệu	1
2	Nguyên lý giải thuật	1
3	Cấu trúc chương trình	1
4	Giải thích mã nguồn C++	2
5	Kết luận	3

1 Giới thiệu

Chương trình được viết bằng ngôn ngữ C++ để tô màu các đỉnh của đồ thị vô hướng sử dụng thuật toán tham lam (Greedy Coloring). Dữ liệu đồ thị được đọc từ file `p2.txt`, chương trình sẽ in ra màu của từng đỉnh và tổng số màu đã sử dụng.

2 Nguyên lý giải thuật

- Khởi tạo màu cho đỉnh đầu tiên là 0.
- Với mỗi đỉnh tiếp theo:
 - Duyệt các đỉnh kề và đánh dấu màu đã dùng.
 - Gán cho đỉnh hiện tại màu nhỏ nhất chưa được dùng.

3 Cấu trúc chương trình

- **neighbor**: Danh sách kề của đồ thị (vector các vector).
- **color**: Mảng lưu màu của mỗi đỉnh, khởi tạo là -1.
- Các hàm chính:

- **readGraph()**: Đọc dữ liệu đồ thị từ file.
- **greedyColoring()**: Thuật toán tô màu.
- **printColoring()**: In kết quả tô màu.

4 Giải thích mã nguồn C++

Listing 1: C++ code tô màu đồ thị

```
#define _CRT_SECURE_NO_WARNINGS
#include <iostream>
#include <vector>
#include <string>
#include <sstream>
#include <algorithm>
using namespace std;

int num_vertex, num_edge;
vector<vector<int>> neighbor;
vector<int> color;

void readGraph() {
    cin >> num_vertex >> num_edge;
    neighbor.resize(num_vertex);
    color.resize(num_vertex, -1);

    cin.ignore();
    for (int i = 0; i < num_vertex; ++i) {
        string line;
        getline(cin, line);
        stringstream ss(line);
        int adjacent;
        while (ss >> adjacent) {
            neighbor[i].push_back(adjacent);
            neighbor[adjacent].push_back(i);
        }
    }
}

void greedyColoring() {
    color[0] = 0;
    for (int u = 1; u < num_vertex; ++u) {
        vector<bool> available(num_vertex, true);
        for (int v : neighbor[u]) {
            if (color[v] != -1) {
                available[color[v]] = false;
            }
        }
        int cr;
        for (cr = 0; cr < num_vertex; ++cr) {
            if (available[cr]) break;
        }
        color[u] = cr;
    }
}
```

```

void printColoring() {
    cout << "Phan_to_mau_do_thi:\n";
    for (int u = 0; u < num_vertex; ++u) {
        cout << "Dinh_" << u << ":_Mau_" << color[u] << "\n";
    }
    int num_colors = *max_element(color.begin(), color.end()) + 1;
    cout << "\nTong_so_mau_su_dung:" << num_colors << "\n";
}

int main() {
    ios_base::sync_with_stdio(0);
    cin.tie(0); cout.tie(0);
    freopen("p2.txt", "r", stdin);
    readGraph();
    greedyColoring();
    printColoring();
    return 0;
}

```

5 Kết luận

Thuật toán tô màu tham lam đơn giản và hiệu quả với đồ thị nhỏ đến vừa. Mặc dù không luôn đảm bảo sử dụng số màu tối thiểu, thuật toán vẫn cho kết quả chấp nhận được trong thời gian ngắn. Chương trình minh họa tốt việc áp dụng thuật toán tham lam vào bài toán tô màu đồ thị.