

Đại học Quốc gia Thành phố Hồ Chí Minh
Trường Đại học Công nghệ Thông tin

Bài tập nhóm 04

Brute Force

CS112 - Phân tích và Thiết kế Thuật toán

Nhóm: 05

Lớp: CS112.N21.KHTN

MSSV: 21520029, 21521845

Ngày 13 tháng 4 năm 2023

Bài tập

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 a 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.)

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

Bài giải

Our solution is based on general graph traversal algorithm, so it could be implemented on both DFS and BFS. In this document, we will only present the problem on DFS algorithm.

We will assign the color of each node in the graph is not important, so let called the colors is 0 and 1 due to we are trying to verify that the graph is bipartite or not.

For the starting node of the traversal procedure, we could assign it to the color 0.

Consider that we are located at node u in the process, how many cases corresponding to node v (v is an adjacent node of u) that could happen?

- If node v has not been visited before, we should move into it and set the color of v different from the color of u .
- If node v has been visited before, v must have the color that different from u , otherwise the whole graph is **not** bipartite. We should stop the procedure now. (1)

If the algorithm is complete without having any (1) cases, the graph is bipartite.

The complexity of the algorithm is as the same as other graph traversal algorithms, $\mathcal{O}(n)$.