Indian Institute of Technology Kharagpur

CS29003: Algorithms Laboratory, Spring 2022

Assignment 9: Graph Traversals

2PM – 5PM 22ND MARCH, 2022

General Instructions (to be followed strictly)

Submit a single C/C++ source file.
Do not use global variables unless you are explicitly instructed so.
Do not use Standard Template Library (STL) of C++.
Use proper indentation in your code and include comments.
Name your file as `<roll_no>_a9.<extn>`
Write your name, roll number, and assignment number at the beginning of your program.

In today's assignment, we solve some problems concerning an undirected graph $G = (V, E)$ using depth-first search (DFS).

Write a function *read_graph* that reads a graph with n vertices and e edges from the user. The vertices of the graph will be numbered $0, 1, \ldots, n-1$. Read the edges as follows. For each $i = 0, 1, \ldots, n$, read the vertices connected to i via an edge, with input -1 indicating end of the list. Use the *adjacency list* representation to store the graph. Solve the following problems on the input graph. Note that the input graph may not be connected.

(a) Suppose that vertices of the input graph represent different classes/lectures and the presence of an edge between two vertices indicates that the two corresponding classes have common students. Suppose that each class runs for 3 hours and there are exactly two 3-hour slots in a day – one in the morning and the other in the afternoon. A class schedule for a day is called *conflict-free* if classes can be scheduled in a way that no student misses any class (s)he has enrolled in. Write a function *exists_schedule* to determine whether or not there exists a conflict-free schedule. Your algorithm must run in $O(n + e)$ time.

(b) *Removal* of a vertex v from G results in a graph H which is similar to G except that it does not contain v and all the edges of G incident on v. A *trivial vertex* is a vertex whose removal does not disconnect the graph (or does not increase the number of disconnected components in the graph). Design an $O(n+e)$-time algorithm that finds all the trivial vertices of G. Write a function *find_trvial* that implements the algorithm and also prints all the trivial vertices.

In the *main()* function,

- Read n, e and call *read_graph*.
- Call *exists_schedule* and print whether or not there exists a conflict-free class schedule.
- Call *find_trivial*.

Do not use any built-in library functions.

- **Sample Output 1**

```
n = 9
e = 10

Reading edges...
0: 6 -1
1: 2 8 -1
2: 1 5 6 7 -1
3: 4 -1
4: 3 5 -1
5: 2 4 7 -1
6: 0 2 8 -1
7: 2 5 -1
8: 1 6 -1

There exists no conflict-free schedule.

The trivial vertices of the graph are:
0 1 3 7 8
```

- **Sample Output 2**

```
n = 8
e = 9

0: 2 3 5 -1
1: 3 -1
2: 0 6 -1
3: 0 1 7 -1
4: 5 -1
5: 0 4 6 7 -1
6: 2 5 -1
7: 3 5 -1

There exists a conflict-free schedule.

Trivial vertices of the graph are:
0 1 2 4 6 7
```

Policy on Plagiarism

Academic integrity is expected from all the students. Ideally, you should work on the assignment/exam consulting only the material we share with you. You are required to properly mention/cite anything else you look at. Any student submitting plagiarised code will be penalised heavily. Repeated violators of our policy will be deregistered from the course. Read this to know what is plagiarism.