Indian Institute of Technology Kharagpur

CS29003: Algorithms Laboratory, Spring 2022

Test 2

2PM – 5PM 5TH APRIL, 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>_t2.<extn>`
Write your name, roll number, and assignment number at the beginning of your program.
Submit on Moodle before the deadline. Submissions by any other means will not be considered for evaluation.
Stay on the MS Team meeting (in our respective channel) throughout, with your video turned on.

Let $G = (V, E)$ be a directed graph with $V = \{v_0, \ldots, v_{n-1}\}$. G is said to be *ordered* if the following hold.

- Every directed edge in G is of the form (v_i, v_j) with $i < j$.
- For every vertex $v_i \in V \setminus \{v_{n-1}\}$, there is atleast one edge leaving v_i.

The *length* of a path is the number of edges in it.

Your task if to find the length of the longest path from v_0 to v_{n-1} and output such a path.

(a) Write a function *read_graph* that reads a graph with n vertices. The vertices of the graph will be numbered $0, 1, \ldots, n-1$. Read the edges as follows. For each $i = 0, 1, \ldots, n-2$: read all vertices j such that G contains edge (i, j), with the user entering -1 to indicate end of the list of vertices. Use the *adjacency list* representation to store the graph.

(b) Write a function *greedy_path* that implements the following greedy strategy. Set $u = v_0$ and $l = 0$. Initialise path p to contain v_0. While there is an edge out of u: choose edge (u, v_j) with j as small as possible; set $u = v_j$ and increment l; add u as the next vertex in p. Print l and p.

(c) The greedy method does not correcly solve the problem. Write down the description of a graph in a commented section following *greedy_path* for which the method does not work. (Description of the graph should contain in line i, for $i = 0, 1, \ldots, n-1$, "i: " followed by list of vertices to which there exists an edge from i). Your example should be different from the one given in the sample output.

(d) Write a function *dp_path* that efficiently solves the problem using dynamic programming (the function must print the length of the path followed by the path itself). Your algorithm must run in time $O(|E|)$ time.

In the *main()* function,

- Read n, the number of vertices. Call *read_graph*.
- Call *greedy_path*.
- Call *dp_path*.

Do not use any built-in library functions.

Sample Output

```
n = 8

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

Greedy:
   Length of the longest path = 4
   Path: 0 -> 1 -> 4 -> 5 -> 7

Dynamic Programming:
   Length of the longest path = 5
   Path: 0 -> 2 -> 3 -> 4 -> 5 -> 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.