

# Coding Assignment 4

## ESO207 2024-25-I

September 23, 2024

### 1 Introduction

In this homework, you will write a C program to implement Dijkstra's algorithm.

*The full marks will be 50. The deadline of this homework is Sunday, September 29, 11 PM IST. Your submissions will not be accepted after this time. Submit a single C file titled '**hw.c**' to the Gradescope active assignment titled '**SSSP**'.*

### 2 Problem Statement

The input will consist of several lines. The first line will have  $n$ ; the set of nodes will be  $\{0, 1, \dots, n-1\}$ . Next  $n$  lines each will have the weighted adjacency list for the nodes. The list for  $i$ -th node will have each child  $j$  of  $i$  followed by the weight of the edge  $(i, j)$  separated by a space. Each adjacency list will end with a -1. Thus if there are no children, the list will only have a -1. The last line will have a single integer denoting the source node. An example input is given below.

```
5
1 2 2 1 -1
0 3 2 9 3 2 -1
3 4 -1
2 6 4 7 -1
0 10 1 5 -1
4
```

Verify that this exactly encodes the graph given in Figure 4 in the "Graph Practice Problems" under resources.

The output will consist of  $n$  pairs integers. Each pair will be of the form  $u v$  denoting that the shortest path from the source will have a cost of  $v$ . The pairs will be sorted on the cost from low to high. So, for the above input, the output will be:

```
4 0 1 5 3 7 0 8 2 9
```

Note that the output ends with a space after the 9 (there is no newline after). Verify that it matches the solution given in the practice problem.

We will make sure that all nodes are reachable from the source so that no distance is  $\infty$ . We'll also make sure there is no tie in the shortest path costs so that the solution is unique.

You may reuse only your code from Coding Assignment 3.

### 3 Test Cases

The above test case will be visible and will carry 10 marks. There will be two hidden test cases each of 20 marks. Thus the total will be 50.

### 4 Submission Instructions

- You must submit a C program titled `'hw.c'`. Other programming languages such as C++, Python, etc. are not allowed. Your code must take the input from “stdin” and write the output to “stdout”.
- Your code will be automatically graded in Gradescope on some test cases as above which will be hidden from you. Therefore, you must make sure that you understand and precisely follow the expected input-output behavior.
- Please write a single C code and name it as `'hw.c'`. This is extremely important. If you violate this, your code will not pass the automatic test cases even if your code runs correctly on your local machine. Common examples of failures include:
  - if you write a C++ program that has the correct input-output behavior
  - if you write two or more different C codes or .h header codes and link them
  - or write a single correct code but name it as `'test.c'`

In any of the above cases, your code will fail. Thus, while you are perfectly allowed to develop your code in your local machine and it works correctly, your code may run into problems in Gradescope until and unless you follow the above instructions.

- Submit your code on *Gradescope* active assignment titled `'SSSP'`. Otherwise your code will not be graded. In particular, do NOT submit on hello IITK or over email. Email to us (instructor or the TAs), meet us, or start a discussion in helloIITK if you run into any issues.