

# COSC2430 Hw4: Path finding from Graph

v1.1

## 1. Introduction

You will create a C++ program that can find path from given graph.

## 2. Input and Output

$G = (V;E)$  is a directed graph, where  $n$  vertices and  $m$  edges.  $G$  can be represented as an adjacency matrix  $E$ ,  $n \times n$ , where  $n \leq 10000$ . Please see Figure 1 as an example. You will read a sparse matrix  $E$  from an input file; There will be ONE matrix entry per line in the input file and each line will have a triplet of numbers  $i, j, \text{length}$  where  $1 \leq i, j \leq n$  indicate the entry,  $\text{length}$  denotes the length for a directed edge pointing from vertex  $i$  to  $j$ . Given a source vertex  $1 \leq s \leq n$  and a destination vertex  $1 \leq d \leq n$ , your program should output the result:

1. The shortest path from  $s$  to  $d$ .
2. The longest path from  $s$  to  $d$ .
3. Total number of paths from  $s$  to  $d$ .

There are two files will be given:

1. Input file: contains the vertices and edges, the form of the input file detailed above. All lengths are decimals and bigger than 0. All vertices are integers and bigger than 0. The file should be read sequentially. If same directed edge with different edge length, only the latter one should be kept. All data will be

meaningful, no error data will be given. Empty lines may be given.

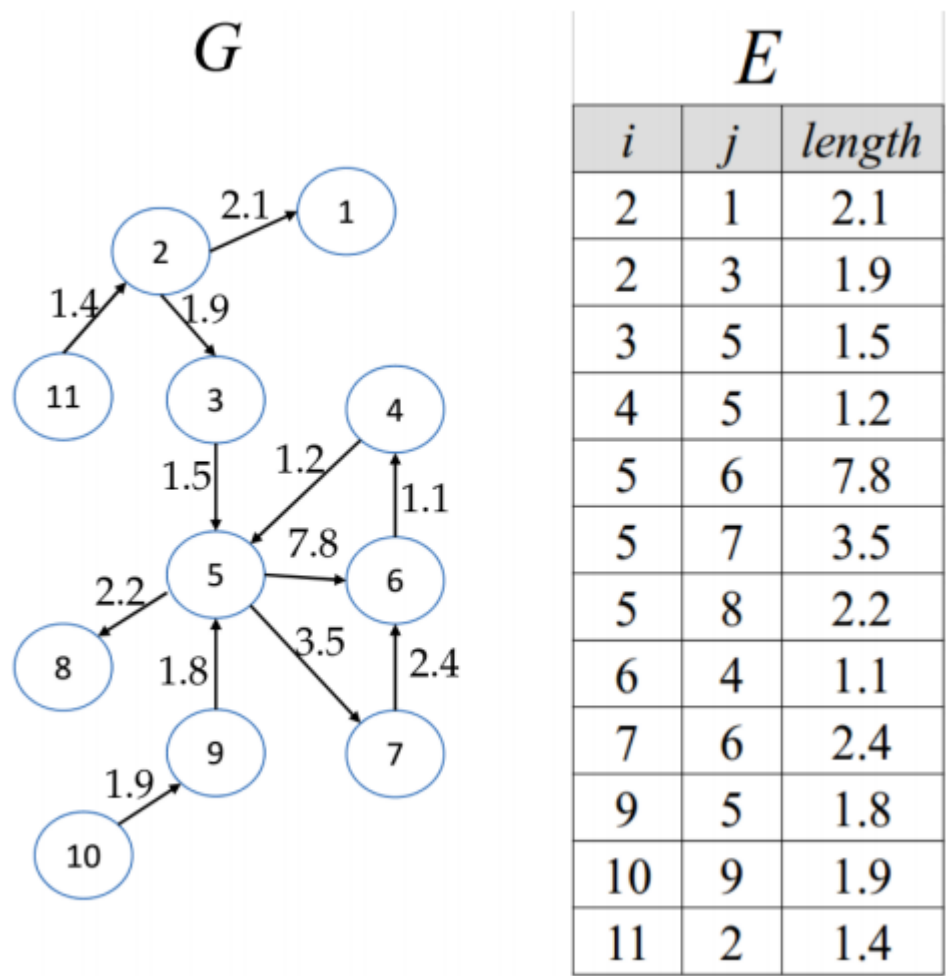


Figure 1: The adjacency matrix  $E$  (sparse representation) for a sample graph  $G$

- 2. **Path file:** contains the number of source and destination vertices. The first line is the source vertex, the second line is the destination vertex. The vertices number may be invalid, if that happens, you should follow the output protocols to output specific result. Only integers will be given, you don't need to consider the letters. Empty lines may be given.
- 3. **Output file:** contains your results. Each result takes one separate line. If two points are unreachable, you should output "Infinite" (case sensitive) for each length searching line, you should output 0 for "total number of path" line. If

one of the endpoints is invalid, you should treat it as unreachable. When you calculate “total number of path”, each vertex only allowed to count once in one path. If the path pass destination vertex, the path should stop at destination vertex. A path starts and ends with same vertex is a 0 length path, the shortest path should be 0.

**Example1 for Figure 1.**

**input41.txt**

```
2 1 1.2
2 3 1.9
3 5 1.5
4 5 1.2
5 6 7.8
5 7 3.5
5 8 2.2
6 4 1.1
7 6 2.4
9 5 1.8
10 9 1.9
11 2 1.4
```

**path41.txt**

```
2
6
```

**output41.txt**

```
9.3
11.2
2
```

**Example2 for Figure 1.**

**input42.txt**

```
2 1 1.2
2 3 1.9
3 5 1.5
4 5 1.2
5 6 7.8
```

5 7 3.5  
5 8 2.2  
6 4 1.1  
7 6 2.4  
9 5 1.8  
10 9 1.9  
11 2 1.4

path42.txt  
11  
7

output42.txt  
8.3  
8.3  
1

Example3 for Figure 1.

input43.txt  
2 1 1.2  
2 3 1.9  
3 5 1.5  
4 5 1.2  
5 6 7.8  
5 7 3.5  
5 8 2.2  
6 4 1.1  
7 6 2.4  
9 5 1.8  
10 9 1.9  
11 2 1.4

path43.txt  
12  
7

output43.txt  
Infinite  
Infinite  
0

### 3. Program and input specification

The main C++ program will become the executable to be tested by the TAs. The result file should be written to another text file (output file), provided with the command line. Notice that the input and output files are specified in the command line, not inside the C++ code. Notice also the quotes in the program call, to avoid Unix/Windows get confused.

The general call to the executable is as follows:

```
graph "input=input41.txt;path=path41.txt;output=output41.txt"
```

Call example with another command line type.

```
graph input=input41.txt path=path41.txt output=output41.txt
```

both type may be used simultaneously.

### 4. Requirements

- Homework is individual. Your homework will be automatically screened for code plagiarism against code from the other students and code from external sources.

If you copy/download source code from the Internet or a book, it is better for you to acknowledge it in your comments, instead of the TAs detecting it. Code that is detected to be copied from another student (for instance, renaming variables, changing for and while loops, changing indentation, etc) will result in "Fail" in the course and being reported to UH upper administration.

- timeout is set to 5s.

## 5. Turn in your homework

Homework 4 need to be turned in to our Linux server, follow the link here

<http://www2.cs.uh.edu/~rizk/homework.html>.

Make sure to create a folder under your root directory, name it hw4 (name need to be lower case), only copy your code to this folder, no testcase or other files needed. If you use ArgumentManager.h, don't forget to turn in it too.

ps. This document may has typos, if you think something illogical, please email TAs or Teacher for confirmation.