COSC2436 Pathfinding from Graph

Created by Kunpeng Zhang (kzhang21@uh.edu)

1. Introduction

You will create a C++ program to find the path from the given graph.

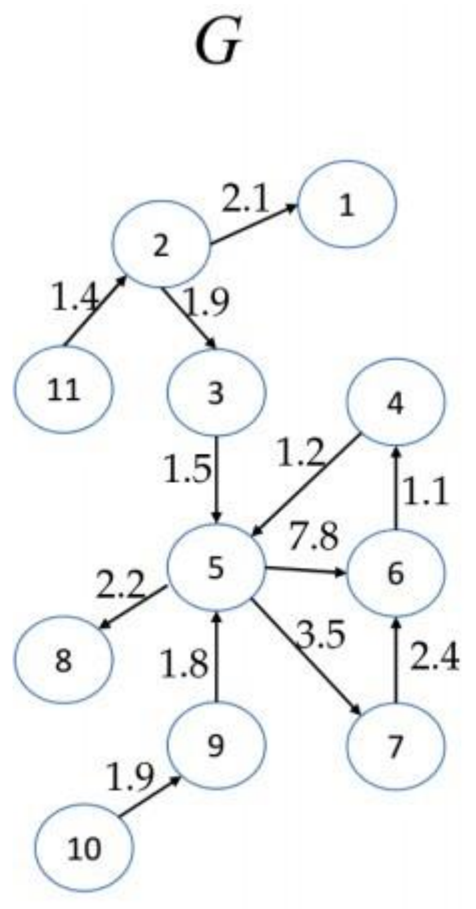
2. Input and Output

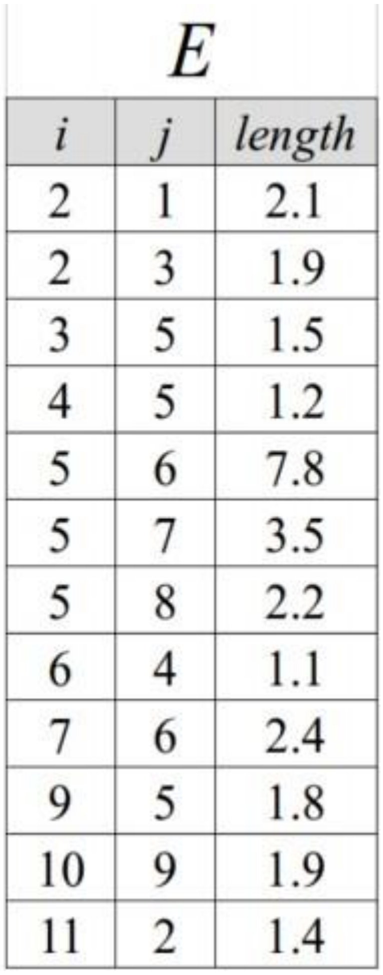
G = (V; E) is a directed graph, where n vertices and m edges. G can be represented as an adjacency matrix E, n x n, where n <= 10000. Please see Figure 1 as an example. You will read a sparse matrix E from an input file; There will be ONE matrix entry per line in the input file, and each line will have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

1. The shortest path value from s to d.
2. The longest path value from s to d.
3. The 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 larger than 0.
    - All vertices are integers and larger than 0.
    - The file should be read sequentially.
    - Only the latter should be kept if the same directed edge has different edge lengths. No error data will be given. Empty lines may be given.





E

i	*j*	*length*
2	1	2.1
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

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

2. Path file: contains the index 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, and if that happens, you should follow the output protocols to output specific results. Only integers will be given. You don’t need to consider the letters. Empty lines may be given.
3. Output file: contains your results.
    - The shortest/longest path value should be the accumulated value from the start vertex to the destination vertex.
    - Each result takes one separate line.
    - If two vertices are unreachable, you should output “Infinite” (case sensitive) for

the shortest/longest value; you should output 0 for the “total number of path” value.

- If one vertex is invalid, you should treat it as unreachable.
- When you calculate the “total number of the path”, each vertex is only allowed to count once in one path (to avoid circles in the path).
- If the path passes the destination vertex, the path should stop at the destination vertex (to avoid circles in the path).
- A path that starts and ends with the same vertex is a 0 length path; the shortest/longest path value should be 0. The “total number of the path” value should be 1.

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

Your source code will be compiled and tested by the TAs. The results should be written to another text file (output file), provided with the command line. Notice the input and output files are specified in the command line, not inside the C++ code. All the necessary parameters will be in the command line when calling your program, so you don't need to worry about the missing parameter problem. When calling your program, the command format would be one of the two standard types as below. Also, notice the quotes in the program call.

The general call to the executable is as follows:

```
graph “input=input51.txt;path=path51.txt;output=output51.txt”
```

Call example with another command line type.

```
graph input=input51.txt path=path51.txt output=output51.txt
```

Both types may be used simultaneously.

4. Turn in your homework

Your program should output results within 2 seconds after will be killed by the system.

Create a folder under your root directory, name it hw5 (name needs to be lower case), and only copy your .cpp and .h files to this folder, no test case.

Homework is individual. Your homework will be automatically screened for code plagiarism against code from the other students and external sources. Code that is copied from another student (for instance, renaming variables, changing for and while loops, changing indentation, etc.) will be detected and result in ”0” in this homework. The limit is 50% similarity. Here is some previous homework found to copy each other (the main function has been deleted).

Ps. This document may have typos; if you think something is illogical, please email TAs for confirmation.