

Programming Lab Assignment 2

Implementing Link-State-Routing Algorithm

(Due: Wednesday, May 04, 2022)

1. Overview

In this programming assignment, you will be implementing Link-State-Routing Algorithm with either C or C++.

1.1 Input File

- The test input file will be randomly generated when your program is tested.
- Test input file format: represent the current status of the entire network
 - The test input file contains link information and the cost of between link
 - $C[i][j] = n$ if there is direct link from node i to node j ,
where $0 \leq n < 100$
 $= 1073741824 (=2^{30})$ otherwise
 - The first line of the test input file has the total number of nodes in the network
 - Following each line contains link cost between i and j .
- Sample test input file:

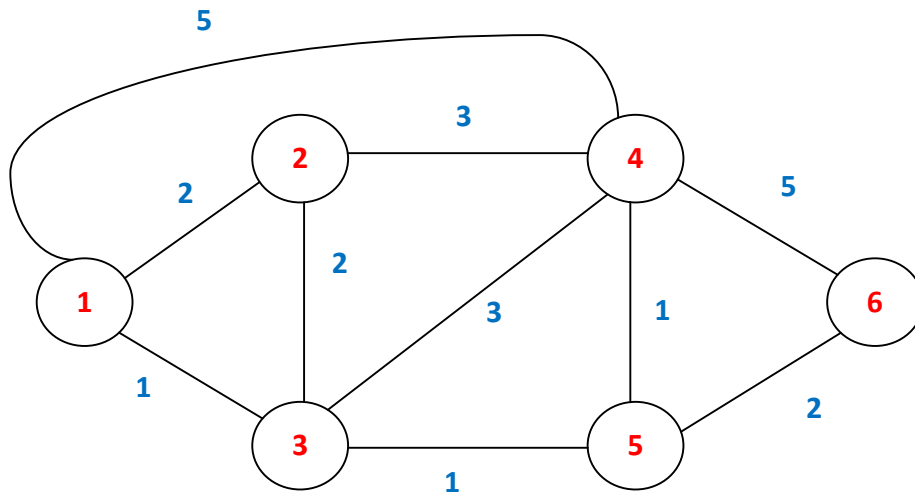


Figure 1: Example Network Topology

Sample test the input file of Figure 1 will be as follows:

```

6
1 1 0
1 2 2
1 3 1

```

```
1 4 5
1 5 1073741824
1 6 1073741824
2 1 2
2 2 0
2 3 2
2 4 3
2 5 1073741824
2 6 1073741824
3 1 1
3 2 2
3 3 0
3 4 3
3 5 1
3 6 1073741824
4 1 5
4 2 3
4 3 3
4 4 0
4 5 1
4 6 5
5 1 1073741824
5 2 1073741824
5 3 1
5 4 1
5 5 0
5 6 2
6 1 1073741824
6 2 1073741824
6 3 1073741824
6 4 5
6 5 2
6 6 0
```

1.2 Your program input arguments

Your program should be able to take two inputs as follows:

Unixprompt> mylinkstate <test-input-file> <node i> <flag>

Where: <test-input-file> : test input file name

<node i> : compute the least-cost paths from <node i> to all possible destination

<flag> : 1 – display immediate table for each iteration

0 – Otherwise (default)

2. Output

2.1 Immediate result

Should be able to display the immediate result for each iteration in a tabular format: (please refer to a table in the class lecture note of chapter 5.1~ 5.2 on page 17- page 20) if the instructor wants to verify the correctness of your program. (when flag value is 1)

2.2 Final result

Your final must display the following two items:

- 1) **Forwarding table for the given <node i>** (Refer to the table in lecture note of chapter 5.1~ 5.2 on page 17- page 20)
- 2) **Total execution time in *ms* (millisecond)** from reading test input files to the completion of computing the least-cost path from the given <node i>

2.3 Program testing

- 1) Your program should run on csegrid without any problems
- 2) You can assume that the maximum number of nodes is 500

3. Grading

The maximum possible point for the assignment is 30. The following criteria will be used for grading this programming assignment.

- Completeness: 20 points
 - If your program works correctly, you will get 20 points
- Completeness of submission: 5 points
- **The efficiency of the program: 5 points max**
 - The top 10% fastest programs earn 5 points
 - The top 20% of fastest programs get 4 points
 - Others get 3 points

4. How to submit

Please do the following when you submit your programming assignment.

- Create a tar file that contains your written source code, makefile, and readme. DO NOT INCLUDE EXECUTABLES AND OBJECT FILES.
- Please use the following convention when you create a tar file
 - First 3 letters of your last name + the last 4 digits of your student ID
 - e.g., If a student's name is "Bill Clinton" and his ID is 999-34-5678, then his tar file name is "cli5678.tar".
- Please upload it to class Canvas by Wednesday, May 04, 2022, by 11:59 pm. after you create the tar file and