CSCI 5311/4311 – Computer Networks

Programming Assignment #2

Due: Soft-copy, Monday, Nov 7th, 2022 by 10:00 AM, via Moodle.

Instructions:

- Before starting the assignment, please review the "integrity" section on the syllabus page. The assignment must be done by you alone.
- **5% overall bonus will be given** when 100% accuracy is achieved for all test cases (checked by the instructor), and each round within a solution is presented (generated) by neat graphics.

Problem Statement:

For a given set of switches/bridges and their connections among each other, determine the spanning tree configuration for the active data path among the switches, computed based on the spanning tree algorithm that we have learned.

To Do:

1. [**Program-Input**] Your program will run and read the inputs from the input.txt file. Each line in the input file is an individual problem.

For example, the first problem is given by input "5 R", which indicates that there are 5 switches and the connections among themselves are randomly (R) determined (by your "rational" program – an example of an "irrational" program will be the one which, may pick all switches with 0 connection most of the time – this will result in a poor grade).

The second line indicates problem #2. Here, a set of individual connections is to be solved among 5 switches given by "5 1-2 1-3 1-3 2-3 3-4 …". Here the first digit "5" implies that there are 5 switches. Then, "1-2" indicates that switch#1 and switch #2 are physically connected by a wire. Note, by "1-3 1-3" it is indicated that switch #1 and switch #3 are physically connected twice: connection #1 is connecting port p2 (of switch #1) and p1 (of switch#3); whereas connection #2 is connecting p3 (of switch #1) and p2 (of switch #3). Here, p2 indicates port number #2, and so on.

The ports of a switch are to be connected/allocated from lower numbered port to higher numbered port for each connection by order of the connection read from left to right for a given problem.

2. Your program will compute the corresponding spanning tree by applying the algorithm discussed in the class (slides). It will draw each round of intermediate configurations as output, including the final spanning tree configured at the end.

3. [**Program-Output**] Your outputs will print the problem configured first (such as "5 1-2 1-3 1-3 2-3 3-4 1-5 …"). Then, it will draw the computed each round of intermediate configurations sequentially. Finally, it will draw the final spanning tree. The whole process is to be done periodically for all the problem(s) given in the input file.

Submission:

(a) In **Report_PA2_<YourID>.doc/x**, provide all different (screen) outputs generated by your program for all the problems given in the input.txt.
(b) Provide the source code and executable. The executable must be easy to run. It must not ask to install a programming package and not ask to compile your code. You can code in java (see the guideline for coding below).
(c) Provide a readme file that will describe how to run your code.
(d) Compress everything in one folder except the report file and submit the compressed folder **PA2_<YourID>.zip** and the **Report_PA2_<YourID>.doc/x**.

Guideline for coding

You will **submit** all your source code in a single java* file called SpanningTree.java. You may develop your code using separate files, but you will need to combine your source code into a single compilable file for submission. The file SpanningTree.java should begin with a public class SpanningTree that contains the main program, which takes the input file name as a command line argument and then performs the simulation of the given switch/bridge configuration problem in the input file. The public class SpanningTree should be followed by your other classes that comprise your system. Recall that java will only allow one class within a source file to be declared public, so your SpanningTree.java source file will look something like this:

```
public class SpanningTree {
        public static void main( String [] args ) {
        … here lies the code to process the input file and solve the Spanning Tree problem …
        }
}
```

Run your program and use the input.txt to get input as:

```
$ java SpanningTree input.txt
```

* If you are using a programming language other than java, please adhere to the naming convention (e.g., SpanningTree.java can be SpanningTree.c)

Outputs (of moderate quality) with possibly correct steps may look like the following:

```
Solution of Problem #1
The number of Switches 5
Input: 5 R
Random Implementation
Printing Initial Connections
**********************************************************************
Source Switch 2 Port Number 1------Destination Switch 1 Port Number 1
Source Switch 2 Port Number 2------Destination Switch 5 Port Number 1
Source Switch 3 Port Number 1------Destination Switch 4 Port Number 1
Source Switch 4 Port Number 2------Destination Switch 2 Port Number 3
Source Switch 1 Port Number 2------Destination Switch 3 Port Number 2
Source Switch 3 Port Number 3------Destination Switch 1 Port Number 3
Source Switch 1 Port Number 4------Destination Switch 5 Port Number 2
Source Switch 5 Port Number 3------Destination Switch 1 Port Number 5
Source Switch 1 Port Number 6------Destination Switch 4 Port Number 3
Source Switch 2 Port Number 4------Destination Switch 1 Port Number 7
**********************************************************************
Initial Status of Switches
```

```
*****************************
SwitchId 1 sends--> 1 1 0
SwitchId 2 sends--> 2 2 0
SwitchId 3 sends--> 3 3 0
SwitchId 4 sends--> 4 4 0
SwitchId 5 sends--> 5 5 0
*****************************
Loop 1
**************
Status of Switches
*****************************
SwitchId 1 sends--> 1 1 0
SwitchId 2 sends--> 2 1 1
SwitchId 3 sends--> 3 1 1
SwitchId 4 sends--> 4 1 1
SwitchId 5 sends--> 5 1 1
*****************************
Printing Final Connections
************************************************************************
Source Switch 2 Port Number 1------Destination Switch 1 Port Number 1
Source Switch 1 Port Number 2------Destination Switch 3 Port Number 2
Source Switch 1 Port Number 6------Destination Switch 4 Port Number 3
Source Switch 5 Port Number 3------Destination Switch 1 Port Number 5
************************************************************************
************************************************************************
```

Solution of Problem #2

```
The number of Switches 5
Input: 5 1-2 1-3 1-3 2-3 3-4 1-5 1-4 2-5
Printing Initial Connections
************************************************************************
Source Switch 1 Port Number 1------Destination Switch 2 Port Number 1
Source Switch 1 Port Number 2------Destination Switch 3 Port Number 1
Source Switch 1 Port Number 3------Destination Switch 3 Port Number 2
Source Switch 2 Port Number 2------Destination Switch 3 Port Number 3
…
…
…
…
************************************************************************
Initial Status of Switches
*****************************
SwitchId 1 sends--> 1 1 0
SwitchId 2 sends--> 2 2 0
SwitchId 3 sends--> 3 3 0
…
…
*****************************
Loop 1
**************
Status of Switches
*****************************
SwitchId 1 sends--> 1 1 0
SwitchId 2 sends--> 2 1 1
SwitchId 3 sends--> 3 1 1
…
…
*****************************
Printing Final Connections
************************************************************************
Source Switch 1 Port Number 1------Destination Switch 2 Port Number 1
…
…
…
…
************************************************************************
************************************************************************
```

Solution of Problem #3

```
The number of Switches 4
Input: 4 1-3 2-3 4-3 2-4
Printing Initial Connections
************************************************************************
Source Switch 1 Port Number 1------Destination Switch 3 Port Number 1
Source Switch 2 Port Number 1------Destination Switch 3 Port Number 2
Source Switch 4 Port Number 1------Destination Switch 3 Port Number 3
```

```
Source Switch 2 Port Number 2------Destination Switch 4 Port Number 2
************************************************************************
Initial Status of Switches
*****************************
SwitchId 1 sends--> 1 1 0
SwitchId 2 sends--> 2 2 0
SwitchId 3 sends--> 3 3 0
SwitchId 4 sends--> 4 4 0
*******************************
Loop 1
*************
Status of Switches
*****************************
SwitchId 1 sends--> 1 1 0
SwitchId 2 sends--> 2 2 0
SwitchId 3 sends--> 3 1 1
SwitchId 4 sends--> 4 2 1
*******************************
Loop 2
*************
Status of Switches
*****************************
SwitchId 1 sends--> 1 1 0
SwitchId 2 sends--> 2 1 2
SwitchId 3 sends--> 3 1 1
SwitchId 4 sends--> 4 1 2
*******************************
Printing Final Connections
************************************************************************
Source Switch 2 Port Number 1------Destination Switch 3 Port Number 2
Source Switch 1 Port Number 1------Destination Switch 3 Port Number 1
Source Switch 4 Port Number 1------Destination Switch 3 Port Number 3
************************************************************************
************************************************************************
```

---- x ----