ICS 340 Programming Project, Deliverable B (50 pts)

Specification:

Start with your Java program "prog340" which implements Deliverable A.

Do a Depth-first search of the directed graph, starting at the node with value "S" (not case sensitive). List the starting and finishing times of each Node, and the class of each Edge (tree edge, forward edge, back edge, cross edge). Make the starting time of the source node time 1. At many points in the search, you may have to choose which node to explore next. Here are the rules to do so:

1. If you have a choice among two or more unexplored Nodes to explore next, there are two cases:
   a. If the values of all Edges are all integers, choose the Edge with the lowest value. If there is a tie, choose the Edge to the Node whose name comes first in lexicographic order. (All node names are unique.)
   b. Otherwise, if the values of the Edges are not all integers (at least one general alphabetical character or non-positive integer), choose the Edge to the Node whose name comes first in lexicographic order.
2. If you have explored as far as possible from the starting node without exploring every node of the graph, continue from the Node whose name comes first lexicographically from among all unexplored Nodes.

Don't Break Existing Functionality

Running Deliverable A on the test files should still work. Actually, it should always work for any test file in any deliverable.

Administrative Details

The "prog340" handout describes the format of the input file for this and all program deliverables.

As will always be the case in this class, the program must be written in Java and must run on the University Windows computer systems. To ensure this I strongly recommend that you:

1. Use only Oracle Java SE, and
2. Test it on the University systems before submission if you have any doubts about its ability to run on the University Windows.
3. As before, try to minimize disruption to the existing codebase, although you will have to do some machinations to deal with Graphs having Edges with positive integral values differently from more general edges.

Output:

Here is sample output for one graph *AB0.txt*.

```
~          val  AAA   BB    C  DDD    E
Alfa         S    ~    >    ~   99  fig
Bravo       67  999  -42    3    x   ==
Charlie      ~    ~    4    ~   yz    9
Delta       4e    3   22    x    ~  !=2
Echo       yes    ~    ~    ~  d>e   33
```

The output for this file should be:

```
Node     Start Time     End Time
Alpha          1            10
Bravo          2            9
Charlie        3            8
Delta          4            7
Echo           5            6

Edge      Type
AAA-BB      T
AAA-DDD     F
AAA-EEE     F
BB-AAA      B
BB-BB       B
BB-C        T
BB-DDD      F
BB-E        F
C-BB        B
C-DDD       T
C-E         F
DDD-AAA     B
DDD-BB      B
DDD-C       B
DDD-E       T
E-DDD       B
E-E         B
```

The output for graph <name>.txt should be written to file <name>_out.txt, and may be written to the console, too.

Submit:

Submit a full Eclipse package to the open Deliverable B submission folder.

Test Files:

There are four test files that will be used for grading: AB0.txt, ABD6.txt, AB2.txt, and AB3.txt. There are also two test files that will not be used for grading, but which might be useful to test with: ABafc.txt and ABnfc.txt. The Test Files are uploaded in a zipped archive, with a README file that describes each test file.

Grading:

40 points for passing the tests correctly. There are four files at 10 points each, plus 3 points for Deliverable A functionality not being broken; it will be tested on one of the files. 3.5 points each respectively for design and documentation.

Due Dates:

The program is due on Friday, March 6th, at 11:59 pm in the D2L “Deliverable B” dropbox.