Erratum – CSCI-561 Homework 1 – Fall 2014

Correction:

1. DFS:

The DFS Example in HW1.pdf had a mistake. It should be:
Log: Andy-Bob-Claire-Alice
Path: Andy-Bob-Claire-Alice

Clarifications and additional hints:

1. Optimality:

"You will need to write a program to implement the following search algorithms, to help *"Andy"* find ~~optimal~~ traversal link(s) to reach *"Alice"*

- Optimality depends on search algorithms. If the search algorithm does not guarantee optimality, your output path does not need to be optimal.

2. More about Tie-breaking:

You should implement the tie-breaking as specified in the homework handout: "When the costs of two or more nodes are equal, you need to make sure these nodes are popped off the search queue in alphabetical order".

Since several students had questions about how this could be done, you may want to consider the following hints for each search as follows. Please note that these are just suggestions of directions for you to explore, if you are wondering how to get started. Depending on your exact implementation, they may or may not work. These hints are provided to help you think about the different approaches you might take to solve the homework problem, but in the end you need to design your own approach that will comply with the specifications of the problem:

a) BFS: When you insert children to the queue, you might want to try to insert them ***in alphabetical order.***

b) DFS: When you insert children to the stack, you might want to try to insert them in REVERSE alphabetical ***order.***

c) UCS: When you pop a child from the priority queue, if there are multiple equivalent choices about which node to pop next, you might want to try to pop a child from the priority queue ***in alphabetical order.***

3. Compiler Setting

For Java users, please make sure your JAVA compiler version is 1.6 on Aludra.