**CIT 2100 AI**

**Search Algorithms Assignment**

**0. Make a copy:** Log into your google account and do File/Make a Copy of this document to have your own copy that you can type in.

If you need help, please come to Zoom office hours on Sunday 4-5 PM at <http://tiny.cc/zoomberyl>.

**1. Search tree for the Farmer Puzzle:**

Try to solve the following problem by drawing a search tree for it and then highlighting the correct path down the tree to the goal solution. Here’s the problem:

A farmer has a wolf, a sheep, and cabbage with him. He comes to a river and finds a small boat that will hold only one other item. He can't leave the wolf alone with the sheep because the wolf will eat the sheep, and he can’t leave the sheep with the cabbage because the sheep will eat the cabbage. How does he get his cargo safely to the other side?

Figure out the solution to this problem by playing it as a game online at <https://www.proprofsgames.com/wolf-sheep-and-cabbage/> (Click play and then one of the items and then go to see him row across. (Hint: it might help to take an item back to the right bank in the boat to keep it safe)).

Now, complete the search tree below for this problem that outlines all the possible moves by typing in on the next page. You can use a shortcut notation that shows who is on each bank of the river and which direction the boat has traveled so <F,W> ← <S,C> means the Farmer and the Wolf are on the left bank and the sheep and the cabbage are on the right bank after the boat traveled to the left. Put \*\*\* by the last node in the solution. I’ve started the search tree below for you. Notice the leftmost and rightmost branches immediately end because something gets eaten.

Farmer, wolf, sheep, cabbage all on right bank of the river

/ | \

Farmer takes wolf Farmer takes sheep Farmer takes cabbage

<F,W><S, C> <F,S>  <W,C> <F,C>  <W,S>

Sheep eats cabbage! | Wolf eats the sheep!

|

Farmer rows back alone

<S> ← <F,W, C>

/ \

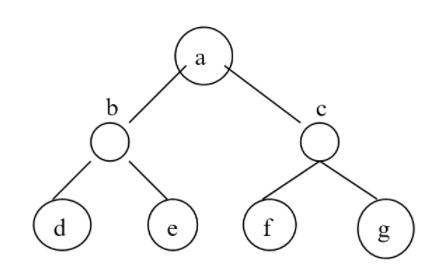
Farmer takes wolf Farmer takes cabbage

<F,S,W> ← <C> <F,W,S> ← <C>

/ \ / \

Complete this tree and mark the goal state!

**2. Search Algorithms:** A robot needs to find its way from room a to room g, but not all the rooms are connected to one another. The following tree shows how the rooms are connected (e.g. a is connected to b and c, but not directly to d (you have to go through room b to get to room d)). The robot must search for the route from room a to room g.



a. If the robot used Breadth-First Search, what order would it visit the rooms before reaching g? List the nodes in the order they are visited starting at a and ending at g following Breadth-first search order. Type in your answer below.

A,B,C,D,E,F,G

b. If the robot used Depth-First Search, what order would it visit the rooms before reaching g? a,b,d,c,e,g,f

c. If the robot used Best First Search using the heuristic of always moving towards the right first, what order would it visit the rooms before reaching g? a,c,f,g

d. If the robot used Best First Search using the heuristic of always moving towards the left first (if you hit a dead-end, backtrack and move right), what order would it visit the rooms before reaching g? A, b, e, d

e. Which search above is the fastest? Is it always guaranteed to be the fastest one (even if you are looking for another node, for example d instead of g)? In my opinion, I would say the best first search is the fastest search out of all the other options.

**3. Coding Challenge:** If you’re a CS major or know Python, look at the following code for BFS and DFS: <https://replit.com/@BerylHoffman/bfs-dfs> . If you’re a CITS major, this is optional. Login with Google at replit.com and click on **Fork** to make your own copy. Change the graph to the one above to check your answers for 2a and b. Include a link to your changed code below.