CS 4700: Foundations of Artificial Intelligence
Spring 2020
Prelim Prep Questions
For Quiz 1

1. True/False Questions:
   a. If the third state visited by both depth-first and breadth-first search is the same (starting from the same initial state with the same goal and using the same tie-breaking strategy) then they must have visited the same second state.
   True. Consider all the graphs that would make some node be the third node visited by depth-first search. Then take the subset for which breadth-first search visits that same node third. All of them visit the same node second.
   b. If you don't check for cycles then breadth-first search can fail to find a solution.
   False. If there's a solution d steps away then breadth-first search will get there after exploring all states that are 1, …, d steps away, including revisiting states that you might have already gotten to multiple times. It won't be as efficient, but it'll still get to a solution.
   c. If iterative deepening finds a solution for a given problem then depth-first search will also find that solution.
   False. Depth-first search might follow an infinite path, but iterative deepening, because of its use of a depth bound, would never follow an infinite path when a solution exists.

2. Consider a state space that is a tree with branching factor b and maximum depth m. Answer parts a and b by placing X's in the appropriate squares in the table below.
   a. Which of the following search methods might take **time** that grows exponentially with m? (Please places X's in the squares in column a corresponding to the search methods for which you think it is true.)
   b. Which of the following search methods might take **space** that grows exponentially with m? (Please places X's in the squares in column b corresponding to the search methods for which you think it is true.)

	a	B
	Can use **time** exponential in m	Can use **space** exponential in m
Depth-first search	X	
Breadth-first search	X	X
Best-first search	X	X

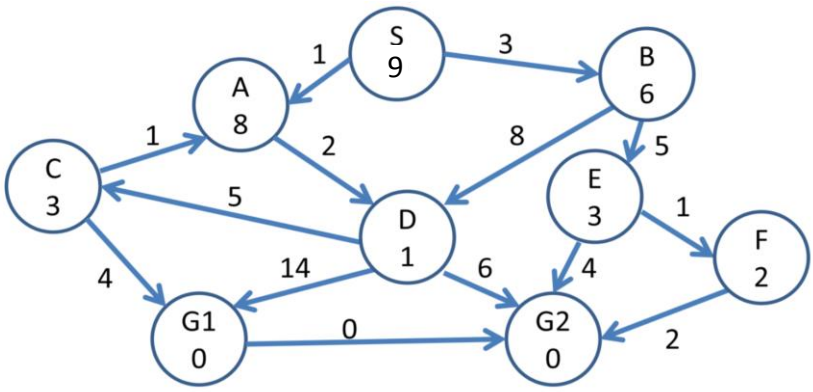
3. For each of the following search methods assume the branching factor is b. Give your answer using big-O notation.
   a. What is the worst-case size of the Open list after depth-first search has visited m states?

The worst-case is when each visited node put b nodes on Open and through the first m nodes we keep generating new nodes. On each iteration b items are put on Open, then one is removed, for a net increase of b-1 items each time. So that means there will be m(b-1) nodes on Open at that point. This is O(bm).

b. What is the worst-case size of the Open list after breadth-first search has visited m states?
Even though breadth-first search can result in Open growing exponentially *with search depth*, that's not what the question asked. The reasoning is the same as for a. If you visit m states, each such visit adds b nodes to Open and then removes 1, so it's also O(bm).

c. What is the worst-case size of the Open list after iterative deepening search has cumulatively visited m states?
Imagine you're on iteration k at the point we've visited m states, so we've done complete searches up to depth k-1. The worst case would be if on level k we've just gone from the root to a node at level k, which would give us O(bk) states on Open. But this is in terms of k, not m. However, since iterative deepening's complexity will have visited $O(b^d)$ if the current iteration goes to depth d, this means k is O(log m). So the answer is O(blog(m)).

5. Consider the following search space, where S is the initial state and G1 and G2 are goal states, where the cost of the operator that takes you from one state to the next is given along the edge between the states, and where the value of the heuristic evaluation function h applied to the state is the number written inside the state. In cases of ties pick the state that comes earlier in the alphabet.



a. Give the sequence of states that depth-first search visits.
S, A, D, C, G1

b. Give the sequence of states that iterative deepening visits.
S, S, A, B, S, A, D, B, E, S, A, D, C, G1