Name: ID:

1. (2 points) The space complexities of BFS and DFS are $O(b^d)$ and $O(bd)$ respectively, where b is the branching factor and d is the search depth. Why is one exponential with respect to d and the other not?

2. (3 points) What is the difference among BFS, DFS, and uniform-cost search (Dijkstra's algorithm) with respect to their implementations in the generic tree search algorithm? (Hint: data structures)

3. (2 points) (a) What is the time complexity of uniform cost search, in terms of the branching factor b and costs C on edges? (b) What is the key assumption that guarantees the optimality of uniform cost search?

4. (2 points) Consider the two statements (a) BFS is a special case of uniform cost search, and (b) uniform cost search is a special case of A*. Under what conditions are they true?

5. (8 points) Sudoku is a popular game in which the player tries to fill in all blank cells so that the resulting board contains 1 to 9 on each row, each column, and each 3 × 3 block.

5	3			7				
6			1	9	5			
	9	8					6	
8				6				3
4			8		3			1
7				2				6
	6					2	8	
			4	1	9			5
				8			7	9

a. (4 points) Formulate it as a tree search problem. What are the state space, goal state, successor function, and the path costs?

b. (2 points) Assume we use uninformed search. Which method would you prefer? Why?

c. (2 points) Is a good heuristic possible in this case? If so, provide one. If not, why not?

6. (8 points) Consider the classic farmer, fox, goose, and grain problem. The farmer wants to move himself, the fox, the goose, and the edible grain from the west side to the east side of the river. Only he can row his small boat across the river, and he can only take one of his items with him at a time. If the fox is left with the goose, the goose will be eaten. If the goose is left with the grain, the grain will be eaten. It turns out that you can pose this as a graph search problem.
   a. (2 points) Describe a representation of the state space for this problem. What would the start state and goal state look like in this representation?

   b. (2 points) What are the possible actions at a particular state?

c. (2 points) Our goal is to get everything to the other side in one piece, so what are the constraints on the state space that we need for the actions in part b?

d. (2 points) Suppose we want to use A* here. Describe a non-trivial admissible heuristic that we could use. Why admissible?