Norman Chung
5550926
CS 165A Homework #3

1a. BFS: 12->10->13->9->11->7->5->8->4->6->14->2->15->16->1->3->17->18->21-> 19->23->20->24 ->25->22->26->27->28->29->30

DFS: 12->10->9->7->5->4->6->2->1->3->21->19->20->22->24->23->25->26->28- >30

IDS:

12

12->10->13

12->10->9->13->11

12->10->9->7->13->11

12->10->9->7->5->8->13->11

12->10->9->7->5->4->8->6->14->13->11

1b. Bidirectional search: 12->30->10->13->28->29->9->11->26->27->7->25->5->8->23->24->4->6 ->14->20- >21->19->22->2->15->16->3->1->3

Two searches meet at state 3

1c. 12->10->13->9->11->7->5->8->4->6->14->2->16->15->18->1->3->17->21->19->23 ->25->20->24 ->27->22->26->29->28->30 output: 12->10->9->7->8->6->2->3->21->23->25->26->28 ->30 cost: 17

2a. a. The state representation is (x, y, z) where x is the number of missionaries that are on the left side of the river, y is the number of cannibals that are also on the left side of the river, and z is the boat position (where 0 means the left side of the river and 1 meaning the right side).

2b. Initial state: (3 3 0), Goal state: (0 0 1)

2c. The successor function in this representation is the set of all valid paths for a given state. So, we can either move 1 or 2 people across the river, while still maintaining the condition that there are more missionaries than cannibals.

2d. The cost function in the successor function is the number of states required, or the number of moves needed to reach the goal.

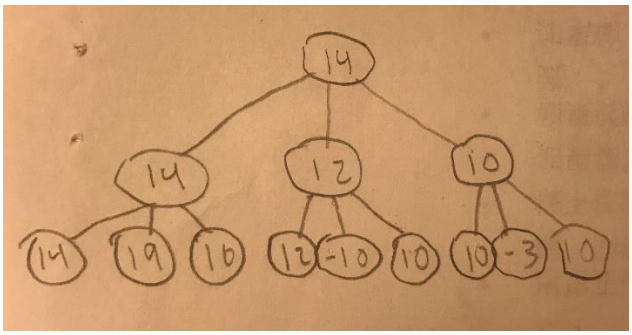
2e. The total number of reachable states is 16, and the total number of unreachable states is 12.

The reachable states are: (3 3 0), (3 2 1), (3 1 1), (2 2 1), (3 2 0), (3 0 1), (3 1 0) , (1 1 1) , (2 2 0), (0 2 1), (0 3 0), (0 1 1), (1 1 0), (0 1 0), (0 2 0), (0 0 1)

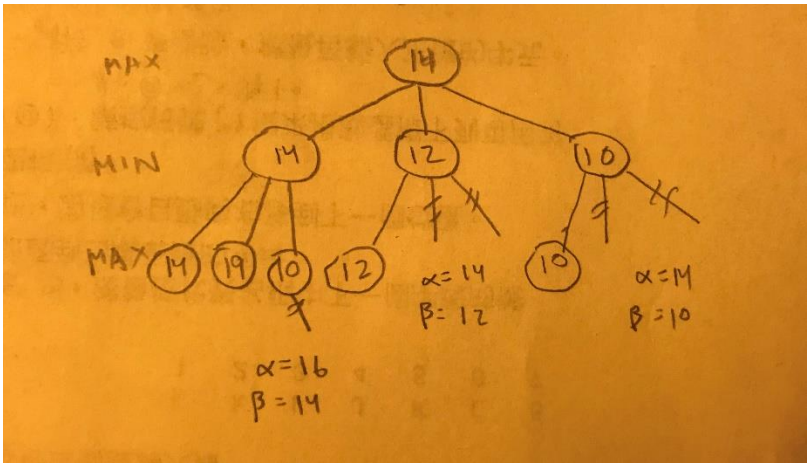
The unreachable states are: (2 3 0), (1 3 0), (2 3 1), (1 2 0), (2 1 0), (2 0 0), (1 3 1), (1 2 1), (2 1 1), (1 0 0), (2 0 1), (1 0 1)

Norman Chung
5550926
CS 165A Homework #3

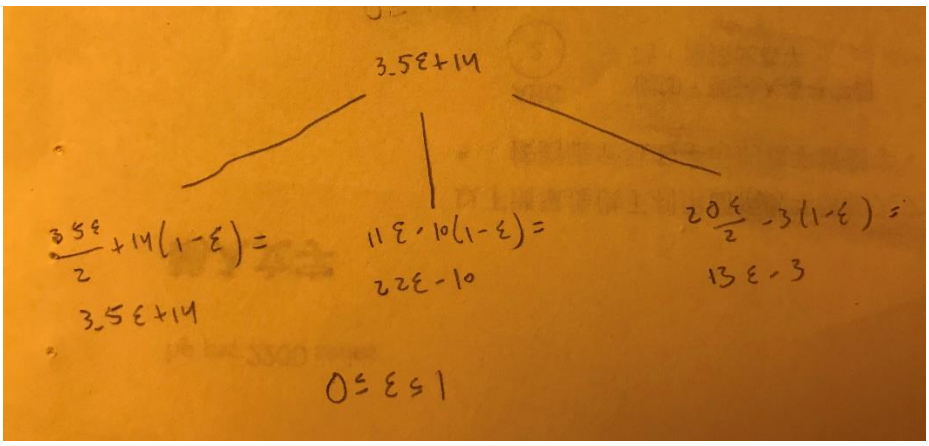
3a.



3b.



3c.



3d. Alpha-beta pruning does not apply in Expectimax. This is because we aren't choosing the minimum value in every iteration, as there is randomness in the choice that the MIN player makes. Thus, we have to check every possible move/choice that MIN can make, and calculate the expected value then.

Norman Chung
5550926
CS 165A Homework #3

4a. Successor function: move the plate to the left or right one space, jump over the plate to the left or right if the destination space is empty. The branching factor is 2.

4b.

CURRENT STATE	EXPANDED NODES	CURRENT PRIORITY QUEUE
(B,B,W,O)	(B,B,O,W) (B,O,W,B)	(B,B,O,W) 1 (B,O,W,B) 2
(B,B,O,W)	(B,O,W,B) (O,B,B,W)	(B,O,W,B) 2 (B,O,B,W) 2 (O,B,B,W) 3
(B,O,W,B)	(O,B,W,B) (B,W,O,B)	(B,O,B,W) 2 (O,B,B,W) 2 (O,B,W,B) 3 (B,W,O,B) 3
(B,O,B,W)	(B,W,B,O)	(O,B,B,W) 2 (O,B,W,B) 3 (B,W,O,B) 3 (B,W,B,O) 4
(O,B,B,W)		(O,B,W,B) 3 (B,W,O,B) 3 (B,W,B,O) 4
(O,B,W,B)	(W,B,O,B)	(B,W,O,B) 3 (B,W,B,O) 4

4c. (b, b, w, o) 0 ->
[b, o, w, b] 2 -> [b, w, o, b] 3 -> [o, w, b, b] 5
[b, o, w, b] 2 -> [o, b, w, b] 3 -> [w, b, o, b] 5

5a. S->A : 1+6=7
S->E : 2+5=7
S->A->B : 5+3=8
S->A->E : 6+5=11
S->E->D : 5+3=8
S->A->B->C : 6+2=8
S->A->B->D : 11+3=14
S->A->B->F : 8+2=10
S->A->B->C->G: 8+0=8

5b. S->A->B->C->G

Norman Chung
5550926
CS 165A Homework #3

6a. i. Initial state: No regions are colored.
ii. Goal state: All regions are colored and no two adjacent regions have the same color.
iii. Successor function: Give an uncolored region a color.
iv. Cost function: Number of color assignments.

6b. i. Initial state: Monkey in initial position in the room.
ii. Goal state: Monkey has the bananas.
iii. Successor function: Moving crates, stacking crates, unstacking crates, grabbing bananas, climbing on crate, climbing off crate, moving on floor
iv. Cost function: Number of actions to get all bananas

6c. i. Initial state: All input records are unprocessed.
ii. Goal state: Finding the record that causes error message.
iii. Successor function: Partition the file until the illegal record is found.
iv. Cost function: Number of partitions.

6d. i. Initial state: Empty jugs.
ii. Goal state: A jug with exactly 1 gallon in it.
iii. Successor function: Fill jugs with water, empty the jugs, pour water from one jug to another.
iv. Cost function: Number of actions needed to get 1 gallon.

7a. The concept of graph search is to traverse and search throughout a graph, and being able to go back to an earlier state in the graph, which is done by keeping track of the states that were previously passed. For an A* graph search algorithm, this refers to the method of using weighted graphs, where you start from a specific node, and find the path with the smallest cost possible, also known as the best first search.

7b. We will prove this with a proof by contradiction. Let us begin by assuming that A* is not an optimal search. This means that whatever path the A* algorithm finds is not the shortest, and is longer than another path. Since we know that the heuristics function is consistent, we know that the estimated cost of reaching the goal is smaller than the actual cost. So, the cost the expand from the starting node must be less than the actual cost. However, for the A* graph search, we know that it always chooses the path with the lowest cost, which makes it the optimal search. Thus, this is a contradiction, and we know that A* is the optimal search.