Allister Liu
ECE-469
HW 1

93/100

1. a) Go
   i) fully observable ✓
   ii) strategic ✓
   iii) sequential ✓
   iv) semi-dynamic if timer ✓
   v) discrete ✓
   vi) multi-agent ✓
   vii) entire percept history ✗ why? without explanation, seems wrong. choice depends on current state of game

23/30

(-1)

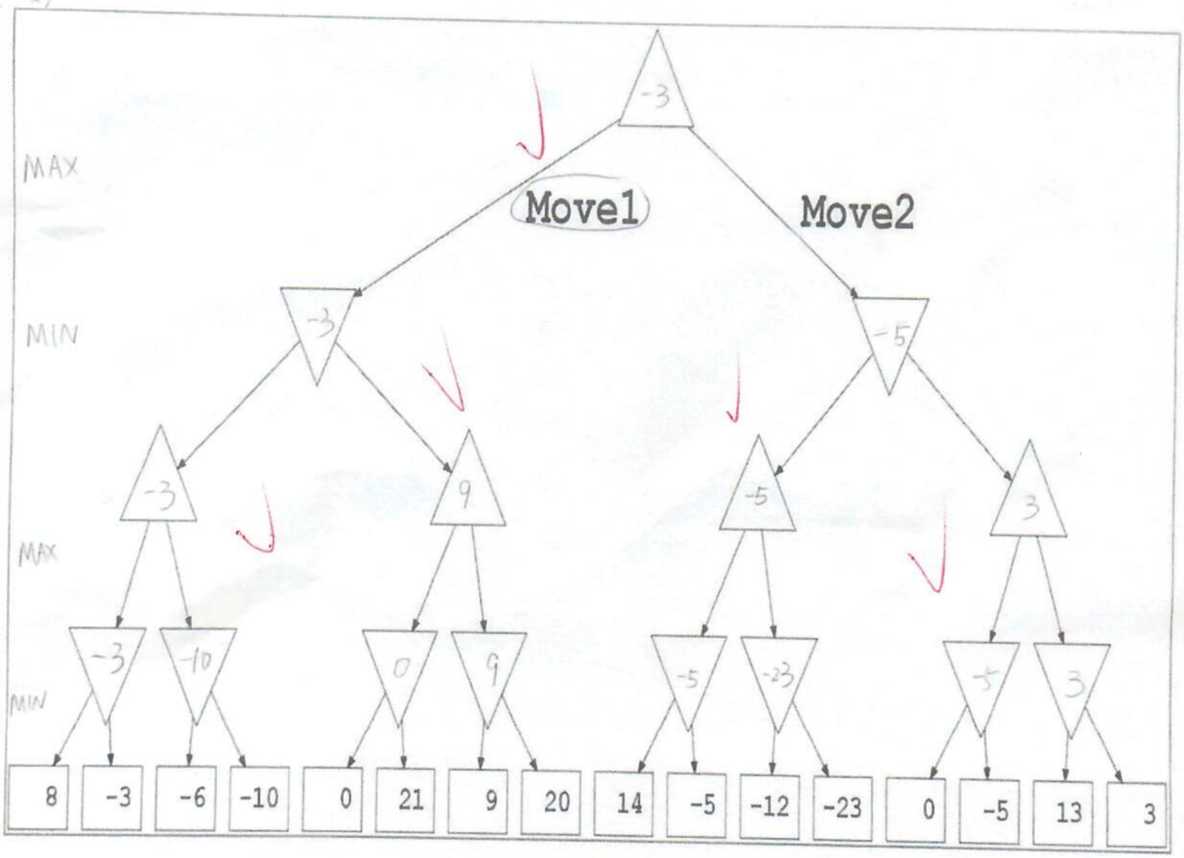
   b) Tenga
   i) partially observable ✓
   ii) deterministic ✗ real world, anything can happen
   iii) sequential ✓
   iv) ~~semi-dynamic~~ evaluation doesn't change while thinking
   v) continuous ✓
   vi) single-agent ✓
   vii) current percept debatable, I'll accept it

(-3)

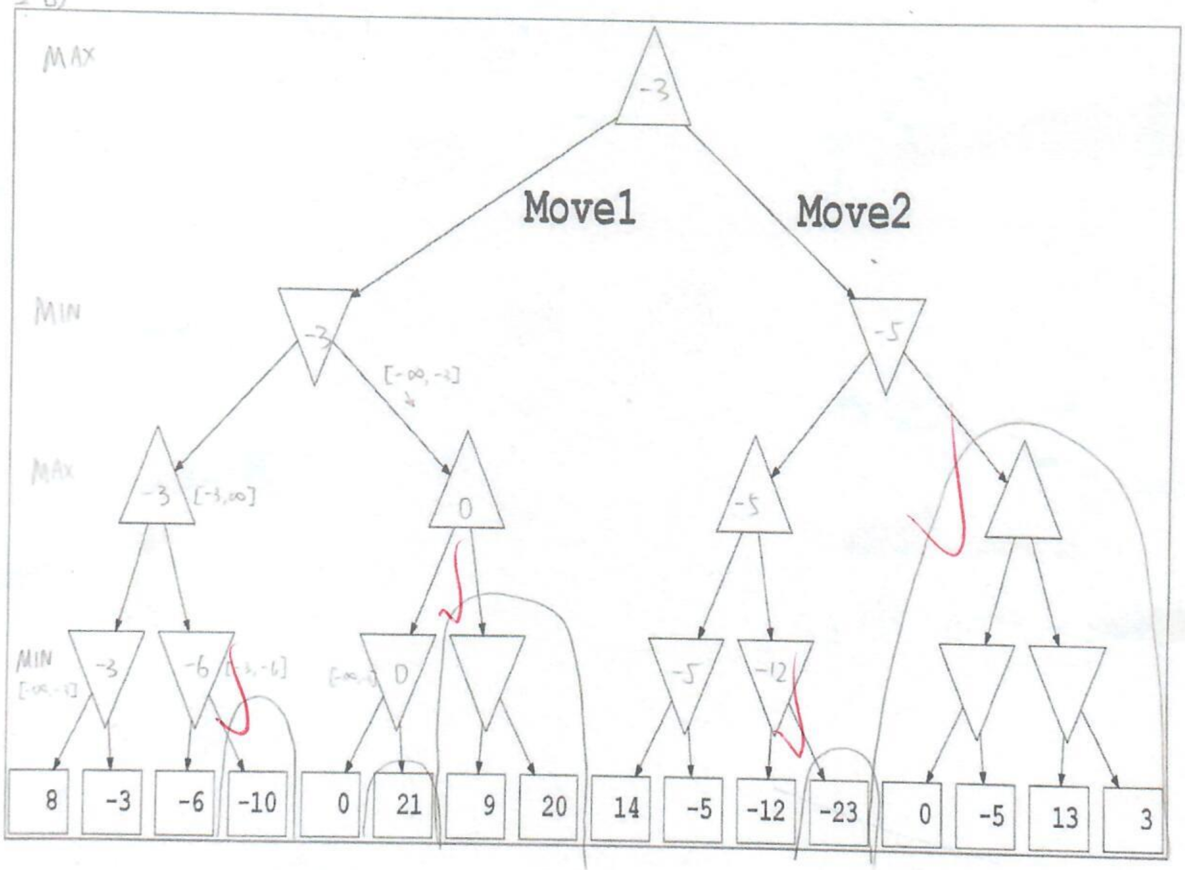
   c) Face recognition
   i) fully observable ✓
   ii) deterministic ✓
   iii) episodic ✓
   iv) semi-dynamic ✗ why?
   v) continuous ✓
   vi) single agent ✓
   vii) entire percept history ✗ image is entire percept, training is done

(-3)

2 a)


MAX
MIN
MAX
MIN
Move1
Move2
-3
-3
-5
-3
9
-5
3
-3
-10
0
9
-5
-23
-5
3
8
-3
-6
-10
0
21
9
20
14
-5
-12
-23
0
-5
13
3

2 b)

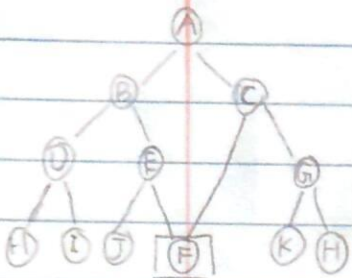

MAX
-3
Move1
Move2
MIN
-3
-5
[-∞, -3]
MAX
-3
[-3,∞]
0
-5
MIN
[-∞, -3]
-3
-6
[-3, -6]
[-∞, -3]
0
-5
-12
8
-3
-6
-10
0
21
9
20
14
-5
-12
-23
0
-5
13
3

30/30

2 c) If both player plays perfectly, MIN will win the game. Because the minimax value is negative, which means if both players play the game perfectly, MIN will always win the game.

40/40

3. a) DFS is complete but not optimal. Because the maze is of size $N \times M$ — finite space, and by remembering nodes on the current path, we can avoid cycles. Therefore DFS is able to perform a complete search.



If a case on the left occurs where A is the enterance and F is the exit, DFS will find A-B-E-F as solution, rather A-C-F which is the optimal solution.

b) BFS is complete and optimal. Because there is a finite branching factor, BFS can perform a complete search. Since the ~~path~~ step cost is constant (path cost is a non-decreasing function of depth), ~~D~~BFS is optimal.

I'm sure that's what you meant

c) H_1, H_2, H_3 would guarantee that the graph search version of A^* search is optimal. It is optimal with a ADMISSIBLE HEURISTIC as long as repeated paths to a state with better costs are not discarded. The shortest (most ideal) path from (x,y) to (N,M) is $N-x+M-y$ (H_2). $H_1 = 0 < H_2$ $H_3 = \sqrt{(N-x)^2+(M-y)^2} < H_2$ by Pythagorean Theorem. $H_4 = (N-x) \times (M-y) > H_2$ ($N \geq x, M \geq y$, x, y, M, N are all positive integers).

d) $H_2(n) = N - x + M - y$ is the best one to use with the graph search version of A^*, because without overestimating, H_2 is the longest (out of H_1, H_2, H_3).

it dominates