Homework 1 EECS 492

1.1 a PEAS

Performance measure: The robots ability to play a legal games of tic tac toe on paper against a human opponent

Environment: The environment is every board played in every game of tic tac toe with information about where the agent plays, human plays and who won

Actuators: multiple motors that allow the robot to move the pen to the correct spot, press down and draw it's shape

SENSORS: Cameras to understand board positions and distance sensor to see how far away we are from the human player & the board

1.1 b [Fully observable] as we always know full board state and have memory of previous games

[Multi-agent]: We play against another agent. so multi-agent

[Deterministic] nothing else changes the board between moves

[Sequential] like chess where our agent plays has consequences so it makes decisions based on more than just an atomic episode

[Static] Assuming no clock our environment does not change while our agent thinks as the board doesn't change

[Discrete] time doesn't factor in, our environment is unchanging during it's turns & possible actions don't change

[Known] assuming we program the rules of tic tac toe into our agent it will be known.

1.2a) BFS expands "row by row" since we use a queue

A → B → H → C → D → I → [J]

GOAL REACHED

1.2b) DFS expand all the way down before recursing up due to it's use of a stack

A → B → C → E → D → F → [G]

goal reached

1.2c) IDS is DFS with set ^max depths that increase as goals aren't found we will end up exploring each level of depth in DFS order. We will have multiple searches at each depth until we find

depth 0 A

depth 1 A → B → H

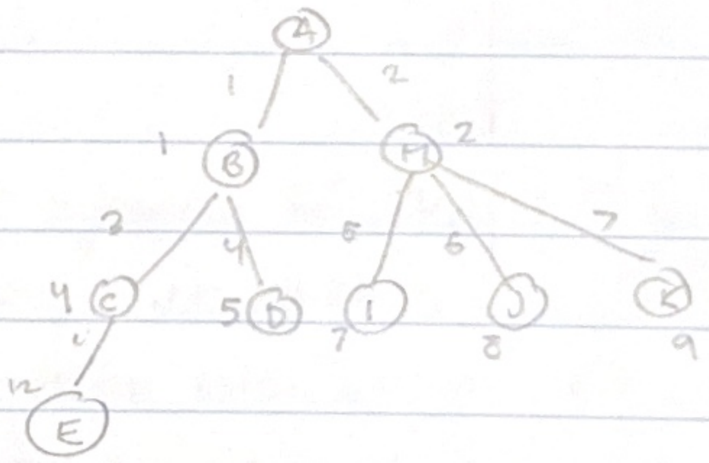
depth 2 A → B → C → D → H → I → [J]

goal reached

1.2d) Uniform cost search uses priority queue & weighted cost so we expand frontier by cheapest cost giving us

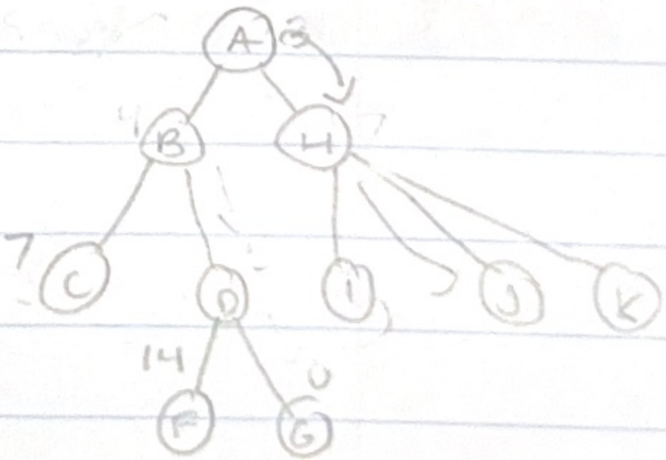
A → B → H → C → D → I → [J]

goal found



1.2e) greedy bfs priority queues our h(n)

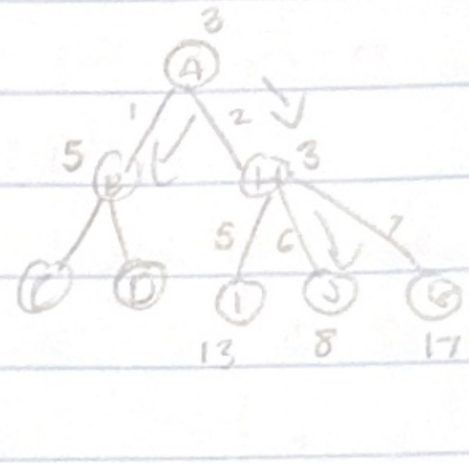
A → H → J



1.2 e) a* also uses a priority queue but sorts values by $g(n) + h(n)$ to force optimality

A → H → B → J

found



1.3 a) You need to know three main things first is the brick mix in all 4 crates, the current position of the robot, if the robot is holding a brick of some type

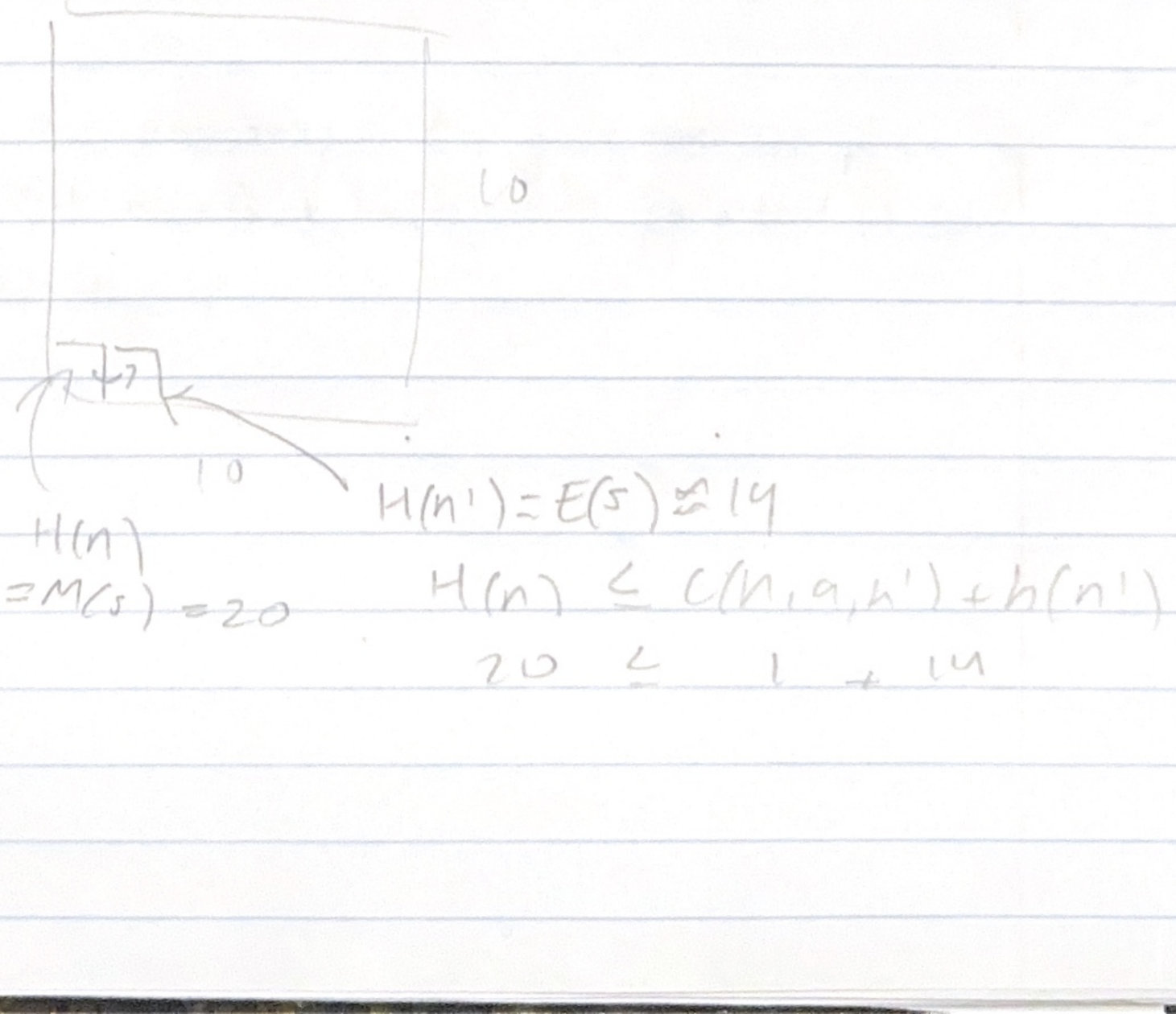
1.3 b) if you move left or right that changes the robot position state, if you pick up a brick that changes the robot brick holding state & the brick mix. The same goes for placing a brick in some crate.

1.3 c) the goal should be defined when the brick mix state for each crate has all the same types of brick and arm holds no brick

1.4 d) Since we are optimizing for shortest sequence of steps and all steps have uniform cost breadth first search is a natural choice. It's optimal for problems with uniform cost steps and is gaurenteed to find a solution at the lowest depth (amount of steps in this case). The runtime is $O(V+E)$ and space is $O(V)$

1.4a) to be consistant $h(n)$ needs to be $\leq c(n,a,n') + h(n')$ take a manhanans distance for a nights move one knights jump away is one while the manhattans distance is 3, thus 3 is greater 1 so it's not consistant. If we change $h(n)$ to be manhatten distance /3 given the furthest we can go in 1 jump (3 rnd \ that would be an $h(n)$ of 1 since $1 \leq 1$ our $h(n)$ is consistant.

14b Since we are only allowed to move up, down, left, and right our distance to goal will equal manhatten distance. Our $h(n)$ has two possible distances 30% chance it's manhatten distance with a 70% it ecclidean. So to prove $h(n) \leq c(n,a,n') + h(n')$ we need $h(n)$ to be decreasing as we get closer. However take the case where we start in the bottom left and our goal is top right 10×10 grid. if we get unlucky our start $h(n)$ is 20 due to being $M(s)$, we move one square towards the goal then our $h(n')$ is in $E(s)$ is about 14, $20 \geq 15$ thus h is not a consistant heuristic.



1.5 a)

PIN	$(x-1, y)$	$(x+1, y)$	$(x, y-1)$	$(x, y+1)$
5,5	0.5	0.416	0.533	0.4
5 4	0.58	0.5	0.7	0.45
5 3	0.75	0.66	1.2	0.53
5 2	1.2	1.16	6.7	6.7
4 4	1.33	1.2	0.75	0.75
3 2	1.5	1.25	0.833	0.83
2,2	2	1.33	1	1
1, 2	-	-	-	-

b) looking at the graph in desmos 3d It's clear that there is one clear peak with no asymetotes so given enough steps we will find the peak of the hill

c) if we have only 10 steps we can consider the worst case scenario. where we are as far as possible from the correct pin 1,1 to 9,9 without loss of generality this would require 16 steps since $1+8, 1+8 = 9,9$, if we only have 10 steps we may not reach the goal in worst cases, However even in worst case with 20 steps we will reach since we only need 16 steps

d) Since there are asymetotes on this graph there are points where we could hit undefined values which could lead us to never reach the pin