ECE 469: Artificial Intelligence
Fall 2020
Problem Set #1

1) Task Environments

In class, we have discussed the following properties of task environments:

i. *fully observable* versus *partially observable*
ii. *deterministic* versus *stochastic* (a third, in-between possibility was *strategic*)
iii. *episodic* versus *sequential*
iv. *static* versus *dynamic* (a third, in-between possibility was *semi-dynamic*)
v. *discrete* versus *continuous*
vi. *single agent* versus *multi-agent*

Note that if you have been reading the relevant textbook chapters, the terms may be slightly different, or the in-between possibilities may be left out, depending on which edition you are using. I am using the terms the way we covered them in class, and the way they appear on my slides. We also talked about whether a task environment is known or unknown (which is not technically a property of the task environment, but of our knowledge of it). I am not asking about that in this question.

Consider the task environments for the following potential AI agents. For each task environment, indicate to which of the above categories the task environment belongs. You can, and should, make certain reasonable simplifications for this question; for example, pseudo-randomness counts as randomness, and digital images should be treated as continuous.

Also, for each task environment, state whether a rational agent would base its decisions on only the *current percept* or the *entire percept history*. So, for each of the potential agents, you are answering *seven things*.

Briefly explain any answers that are not obvious. (There may be answers that are debatable, and for those, I'll give you credit when you disagree with me if your explanation is reasonable. For many parts, I think there is just one clear answer.)

(a) A program that plays Go against the user. (If you need to look up the rules, look them up. Note that the specific algorithm a program uses to play should not affect the answer; this question is about the task environment, not the agent itself.)

i. Fully observable: Can observe the entire board.
ii. Strategic: Aware of how actions will affect state, but unaware of what the other agent will do.
iii. Sequential: Current decision will affect future decisions.
iv. Static: Environment does not change while player is deliberating. If played with a clock, semi-dynamic.
v. Discrete: Finite (but large) amount of states. Excludes the clock.
vi. Multi-agent: Played with 2 agents.
vii. Current percept: Can make a decision solely based on current environment.

(b) A program controlling a robotic arm that detects and stacks Jenga pieces (one of the senior projects this year!). The goal is not to have the robotic arm play the game, but rather it will detect pieces on a surface and build the initial tower.

i. Partially observable: Cannot fully model a continuous environment with discrete sensors. Outside world is extremely complicated, and sensors may be noisy.
ii. Stochastic: Outside world is deterministic, but inaccurate/imprecise sensors require treating it as stochastic.
iii. Sequential: Current decision will affect future decisions.
iv. Dynamic: Environment can change will deciding next action.
v. Continuous: Infinitely many possible states.
vi. Single agent: Only robotic arm is acting on the environment.
vii. Entire percept: Agent probably has an internal model of the world. As the agent performs actions, it will update that model by comparing its model (after taking an action) with the measurements from its sensors.

(c) A face recognition system that detects faces in an image and looks them up in a database of known faces to find matches (assume the system has already been trained).

i. Partially observable: Images may be blurry, and if system uses sensors, sensors will have noise. If quality of image is not considered part of the environment, then the environment is fully observable.
ii. Deterministic: There is no random element involved. Classifying the image is completely deterministic.
iii. Episodic: Current decision does not affect future decisions.
iv. Static: Environment does not change over time.
v. Continuous: Digital images should be treated as continuous.
vi. Single agent: Only system is acting on the environment.
vii. Current percept: Can make a decision solely based on current environment.

ECE 469: Artificial Intelligence
Fall 2020
Problem Set #1

2) Games

Assume a program is playing a deterministic (really strategic), turn-taking, two-player, zero-sum game of perfect information. The game works as follows. A full game takes four moves. MAX takes a turn, then MIN takes a turn, then MAX, then MIN. Then the game is over, scored according to an objective function. Positive values represent wins for MAX (higher is better), whereas negative values represent wins for MIN (lower is better). A game tree for the game is depicted on the following page. MAX nodes are represented as upward pointing triangles, MIN nodes are represented as downward pointing triangles, and terminal nodes are represented as rectangles with the final scores indicated inside the rectangles.

(a) *Assuming that no pruning occurs*, fill in the minimax values of all the internal nodes (including the root) in the game tree. Also circle the move (Move1 or Move2) that is selected according to the search.

(b) Now *assuming that alpha-beta pruning occurs*, draw lines through the edges in the game tree that separate the nodes at which the pruning decision occurs from the nodes which are never considered. (In other words, the node above each line will be a node at which a pruning decision occurs after the evaluation of one or more of its children, and the subtree below each line will contain nodes that are not considered at all due to the pruning.) HINT: You should NOT have to step through the entire alpha-beta search algorithm to do this. That would take longer than necessary. If you understand conceptually what alpha-beta pruning does, you should be able to carefully eyeball the graph and figure this out.

(c) Answer below: If both players play perfectly, what will be the outcome of the game (i.e., will MAX or MIN win or will it be a draw)? Briefly explain your answer.

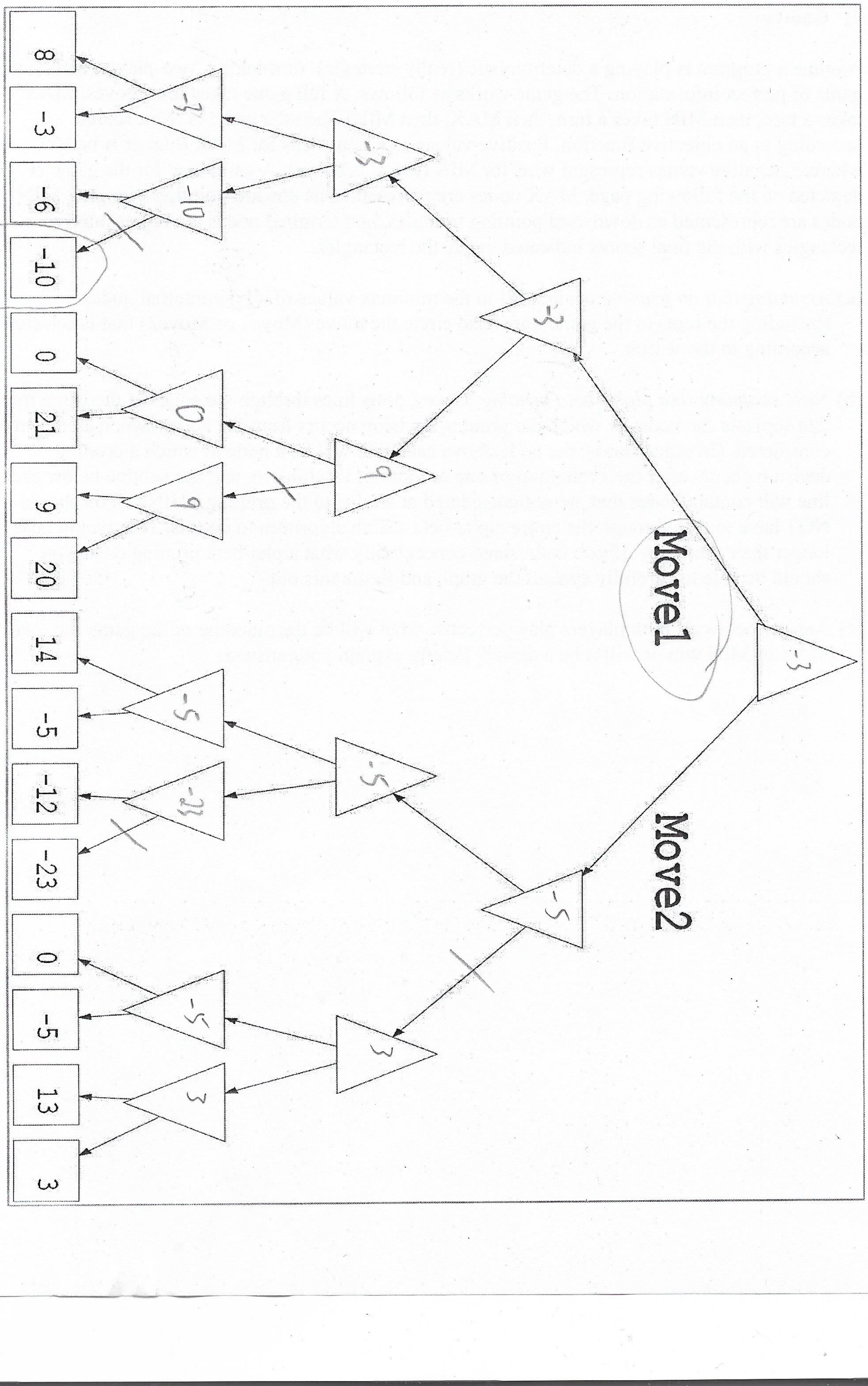
MIN will win. Assuming both players play optimally, the final result will be -3.

ECE 469: Artificial Intelligence

Fall 2020

Problem Set #1

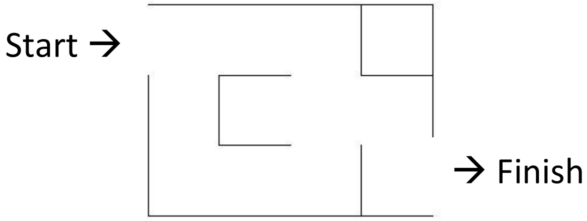
min max min max



ECE 469: Artificial Intelligence
Fall 2020
Problem Set #1

3) Search Strategies

Consider an AI software agent (a program) that processes a maze contained in an N x M grid, where N (the number of rows) and M (the number of columns) are guaranteed to be at least 2. The entire maze is available to the agent at once, not as an image, but as an array indicating where the walls are. The agent is tasked with finding a path from the top left square to the bottom right square. The path cost is the number of steps (i.e., every move from one square to an adjacent square adds one to the cost). Only horizontal and vertical steps are allowed. Walls will only occur between squares. It is possible that there may be loops, multiple paths to the solution, or perhaps no path to the solution. An example of such a maze, where N is 3 and M is 4:



Now answer the following question. (Do not assume the specific maze shown here, it is just an example; N, M, and the maze will be provided to the agent.) *Briefly explain all your answers.*

(a) First consider applying a tree search version of depth-first search (DFS) to this problem. For this question, consider a version that does not remember all reached nodes, but that does remember nodes on the current path to avoid cycles. Is this strategy complete? Is this strategy optimal?

Yes, this strategy is complete, assuming N and M are less than infinity. It should be a reasonable assumption, because it would be impossible to get to the bottom right (and thus have no solution) if N or M are infinity. However, it is not optimal. It is possible to contrive a problem such that DFS finds a suboptimal solution.

(b) Now consider applying a graph search version of breadth-first search (BFS) to this problem, as discussed in class. Is such a strategy complete? Is such a strategy optimal?

Yes, this strategy is complete. If a solution exists, BFS will find it. Yes, this solution is optimal. It will always find the best solution if it exists, given infinite memory and time.

ECE 469: Artificial Intelligence

Fall 2020

Problem Set #1

(c) Now consider applying the graph search version of A* search, as discussed in class, to the problem. You will consider four heuristic, where x is the current row and y is the current column. Assume rows are labeled from 1 to N and columns are labeled from 1 to M; the top-left cell (the starting position) is (1, 1), and the bottom-right cell (the destination) is (N, M). The four heuristics we are considering are:

- $H_1(n) = 0$
- $H_2(n) = N – x + M - y.$
- $H_3(n) = [(N – x)^2 + (M – y)^2]^{½}$
- $H_4(n) = (N – x) * (M – y)$

Which of the above four heuristics would guarantee that the graph search version of A* search is optimal? Specify all that apply.

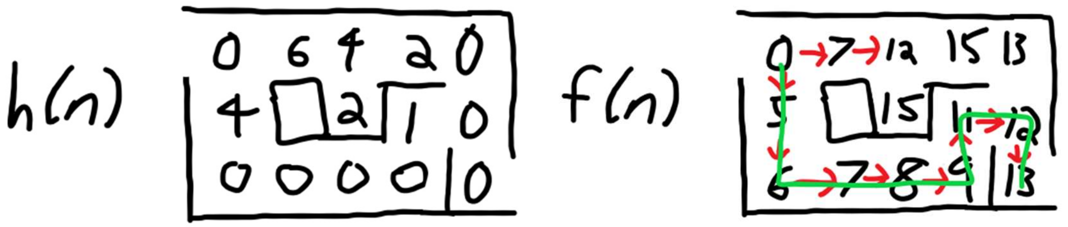
i. H1: Always 0. A* turns into Uniform-Cost Search, which never overestimates the shortest path to the goal.

ii. H2: Manhattan distance; Will never overestimate the distance required because it is the shortest path to the goal assuming there are no walls.

iii. H3: Euclidean distance; Will never overestimate the distance required because it is theoretically the shortest path to the goal (but it is not, because only horizontal and vertical steps are allowed).

H4: Cannot guarantee that A* is optimal. It can find a suboptimal solution.

E.g.



(d) Which of the above heuristics is the best one to use with the graph search version of A* search? Why?

i. H2: It will always dominate H1 because $H2 \geq 0$. It will always dominate H3 because of the triangle inequality. H4 can find a suboptimal solution, so it would not be considered.