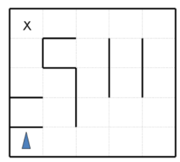
CS 188
Spring 2019

Section Handout 2

1 Search and Heuristics

Imagine a car-like agent wishes to exit a maze like the one shown below:



The agent is directional and at all times faces some direction $d \in (N, S, E, W)$. With a single action, the agent can *either* move forward at an adjustable velocity v *or* turn. The turning actions are *left* and *right*, which change the agent's direction by 90 degrees. Turning is only permitted when the velocity is zero (and leaves it at zero). The moving actions are *fast* and *slow*. *Fast* increments the velocity by 1 and *slow* decrements the velocity by 1; in both cases the agent then moves a number of squares equal to its NEW adjusted velocity (see example below). A consequence of this formulation is that it is impossible for the agent to move in the same nonzero velocity for two consecutive timesteps. Any action that would result in a collision with a wall crashes the agent and is illegal. Any action that would reduce v below 0 or above a maximum speed $V_{\max}$ is also illegal. The agent's goal is to find a plan which parks it (stationary) on the exit square using as few actions (time steps) as possible.

As an example: if at timestep t the agent's current velocity is 2, by taking the *fast* action, the agent first increases the velocity to 3 and move 3 squares forward, such that at timestep $t + 1$ the agent's current velocity will be 3 and will be 3 squares away from where it was at timestep t. If instead the agent takes the *slow* action, it first decreases its velocity to 1 and then moves 1 square forward, such that at timestep $t + 1$ the agent's current velocity will be 1 and will be 1 squares away from where it was at timestep t. If, with an instantaneous velocity of 1 at timestep $t + 1$, it takes the slow action again, the agent's current velocity will become 0, and it will not move at timestep $t + 1$. Then at timestep $t + 2$, it will be free to turn if it wishes. Note that the agent could not have turned at timestep $t + 1$ when it had a current velocity of 1, because it has to be stationary to turn.

(a) If the grid is M by N, what is the size of the state space? Justify your answer. You should assume that all configurations are reachable from the start state.

(b) Is the Manhattan distance from the agent's location to the exit's location admissible? Why or why not?

(c) State and justify a non-trivial admissible heuristic for this problem which is not the Manhattan distance to the exit.

(d) If we used an inadmissible heuristic in A* graph search, would the search be complete? Would it be optimal?

(e) If we used an *admissible* heuristic in A* graph search, is it guaranteed to return an optimal solution? What if the heuristic was consistent? What if we were using A* tree search instead of A* graph search?

(f) Give a general advantage that an inadmissible heuristic might have over an admissible one.

Q2. Heuristics

For parts a and b below, consider the following PacMan problem in a $N \times M$ grid, where $N \geq 2$, $M \geq 2$: there is a food pellet located at each corner, and Pacman must navigate the maze to find each one. Our goal is to find the shortest path that eats all four pellets in each corner.

(a) For each of the following heuristics, say whether or not it is admissible. If a heuristic is inadmissible, give a concrete counterexample (i.e., draw a maze configuration in which the value of the heuristic exceeds the true cost).

- h_1 is the maze distance to the nearest food pellet (if no food pellets remain, $h_1 = 0$).

  [*Admissible* or *Not-Admissible*]

- h_2 is the number of uneaten food pellets remaining.

  [*Admissible* or *Not-Admissible*]

- $h_3 = h_1 + h_2$

  [*Admissible* or *Not-Admissible*]

- $h_4 = \max\{h_1, h_2\}$

  [*Admissible* or *Not-Admissible*]

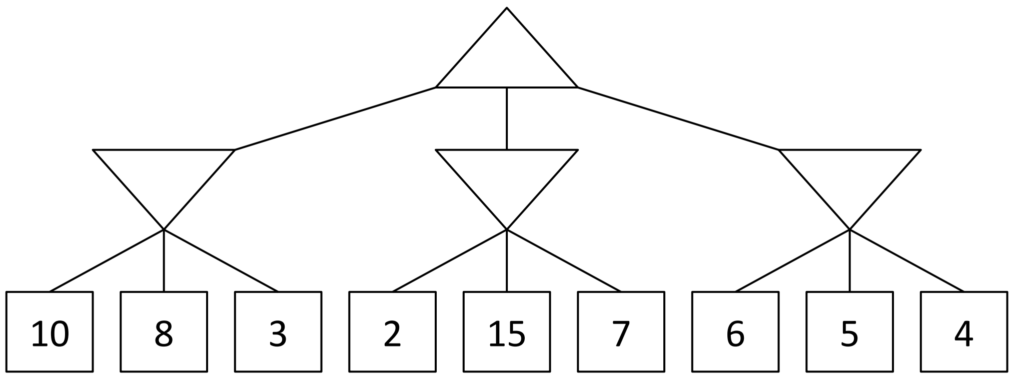
- $h_5 = |h_1 - h_2|$

  [*Admissible* or *Not-Admissible*]

(b) Pick one heuristic from part (a) that you said was inadmissible; call it h_k. Give the smallest integer constant $\epsilon > 0$ such that $h' = h_k - \epsilon$ is an admissible heuristic. Briefly justify your answer.

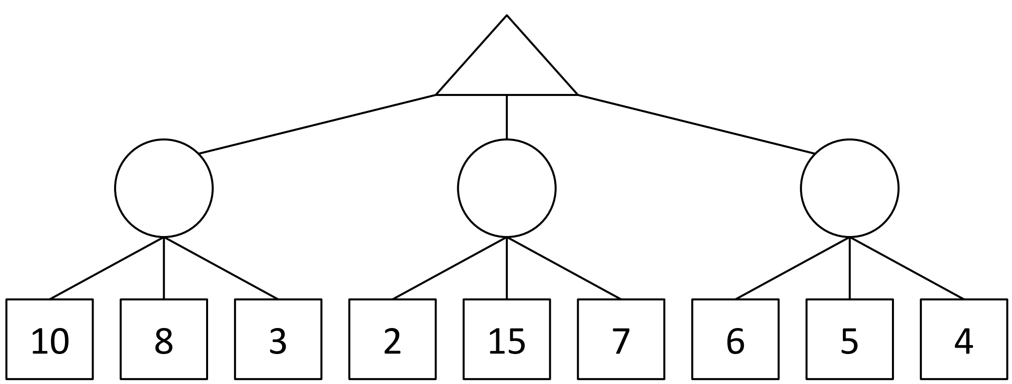
3 Games

(a) Consider the zero-sum game tree shown below. Triangles that point up, such as at the top node (root), represent choices for the maximizing player; triangles that point down represent choices for the minimizing player. Assuming both players act optimally, fill in the minimax value of each node.



(b) Which nodes can be pruned from the game tree above through alpha-beta pruning? If no nodes can be pruned, explain why not. Assume the search goes from left to right; when choosing which child to visit first, choose the left-most unvisited child.

(c) (optional) Again, consider the same zero-sum game tree, except that now, instead of a minimizing player, we have a chance node that will select one of the three values uniformly at random. Fill in the expectimax value of each node. The game tree is redrawn below for your convenience.



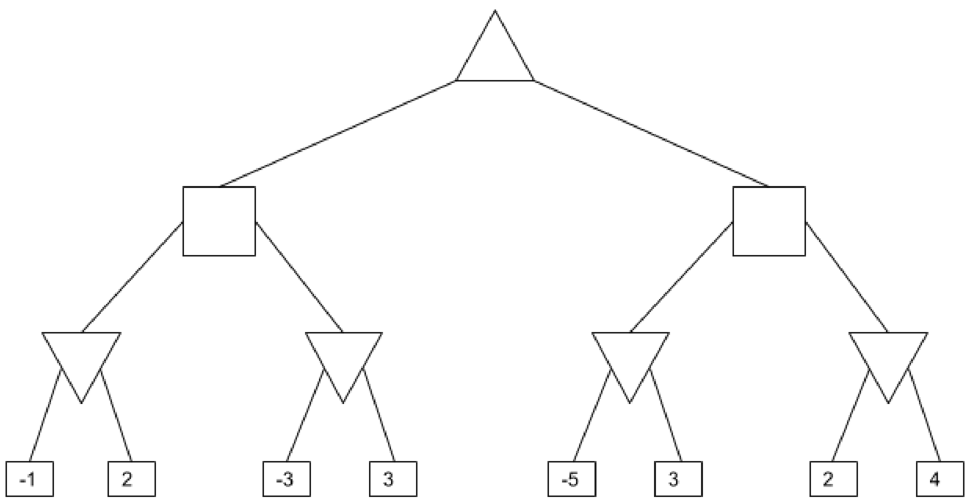
(d) (optional) Which nodes can be pruned from the game tree above through alpha-beta pruning? If no nodes can be pruned, explain why not.

Q4. Fair Play

Consider a game tree with three agents: a maximizer, a minimizer, and an equalizer. The maximizer chooses the highest score, the minimizer chooses the lowest score, and the equalizer chooses tries to *minimize the absolute value* (i.e. equalizer wants to make the game as close as possible, so it chooses whichever value is closest to zero).

We use an upward-facing triangle to represent a max node, a downward-facing triangle to represent a min node, and a square to represent an equalizer node.

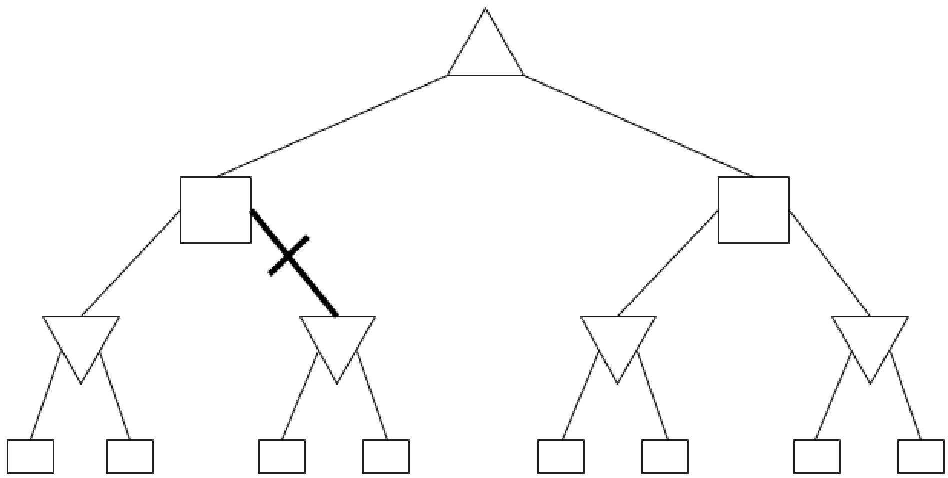
(a) Fill in all values in the game tree below:



(b) In the same game tree above, put an X on the line of all branches that can be pruned, or write "No pruning possible." Assume that branches are explored from left to right.

(c) For each of the following game trees, fill in values in the leaf nodes such that only the marked, bold branches can be pruned. Assume that branches are explored from left to right. If no values will allow the indicated nodes to be pruned, write "Not possible." **Be very clear**: if you write "Not possible," we will not look at the values you filled in.

(i) [Hint: what is the *best possible value* from the equalizer's viewpoint?]



(ii) Note that the order of the players has changed in the game tree below.