**Introduction To Artificial Intelligence**

**HW 2**

Maroon Ayoub – 315085597

Yara Kassem - 212108880

**Part 1**

1. Definitions:

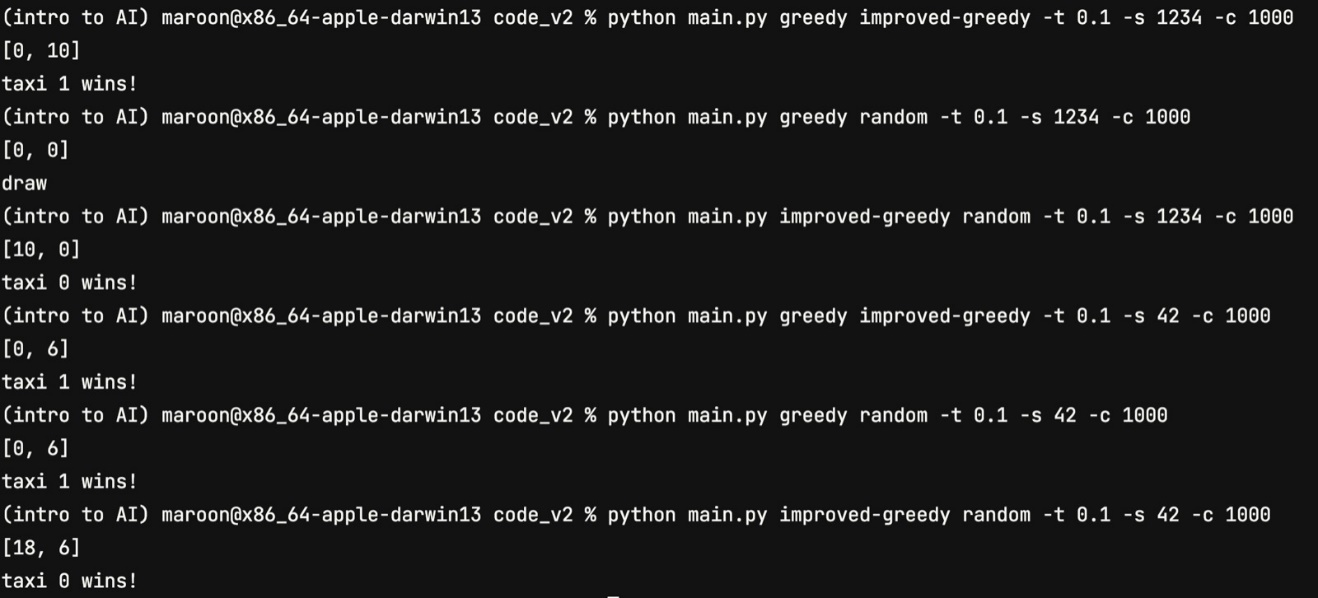
mhs(taxi, passenger) = MD(taxi\_location, passenger\_location) + MD(passenger\_location ,passenger\_destination)

= – other\_taxi\_cash

There are two passengers, the first passenger will be named and the second passenger will be named .

A = mhs( taxi, ) , B = mhs( taxi, , C = mhs( other\_taxi , , D = mhs(other\_taxi ,

1. The improved heuristic guides the agent towards the closest passenger that it is **guaranteed** to reach\* (with the opponent’s distance-to-passengers insight).  
   Once a passenger is reached, it prefers pickups due to the design (state after pickup necessarily has a lower heuristic value than all other sibling states).  
   After a passenger is on board, the heuristic returns the MD between the agent and the dropoff destination. Finally, a dropoff is guaranteed due to the weight given to the agent’s cash.  
     
   \* more accurately - pseudo-guaranteed: the agent competes with opponent if both maintain equal distance to closest passenger, winner depends on action scheduling.



**Part 2**

1. It is not necessarily a bug. The winning move could be on a path in which the enemy does not operate optimally, therefore minimax might consider a different action since it calculates the best of the worst options.
2. Alpha-Beta advantages:

* The number of closed vertices when the alpha-beta method is used is less than the number of closed vertices when other methods are used, resulting in less run- time.
* Since no “unnecessary” vertices are closed, a deeper search is possible compared to other methods (under time constraints) resulting in increased solution quality.
* If there are no time constraints, the alpha-beta method always returns the optimal solution.

Iterative deepening advantages:

* A solution is returned anytime the search is stopped.
* If a good heuristic is given, a good solution can be found during short time.

Possible improvements:

* Combining the two methods above- anytime alpha-beta.
* Reorder the sons in the tree whenever the alpha-beta method is used. The sons of a max vertex are ordered in a descending order, and the sons of a min vertex are ordered in an ascending order.
* Selective deepening when the alpha-beta method is used.

**Part 3**

2. If there were k taxis instead of 2, then the Turn function needs to accommodate the k taxis and switch between them in order.

Runtime grows linearly with k.

3. Regarding runtime, it is possible that the alpha-beta agent’s runtime is less than the minimax agent’s runtime. since less vertices are visited when the alpha-beta method is used compared to the minimax method , it is possible that the alpha-beta agent’s runtime is less than the time limit while the minimax agent’s runtime is equal or longer than the time limit.

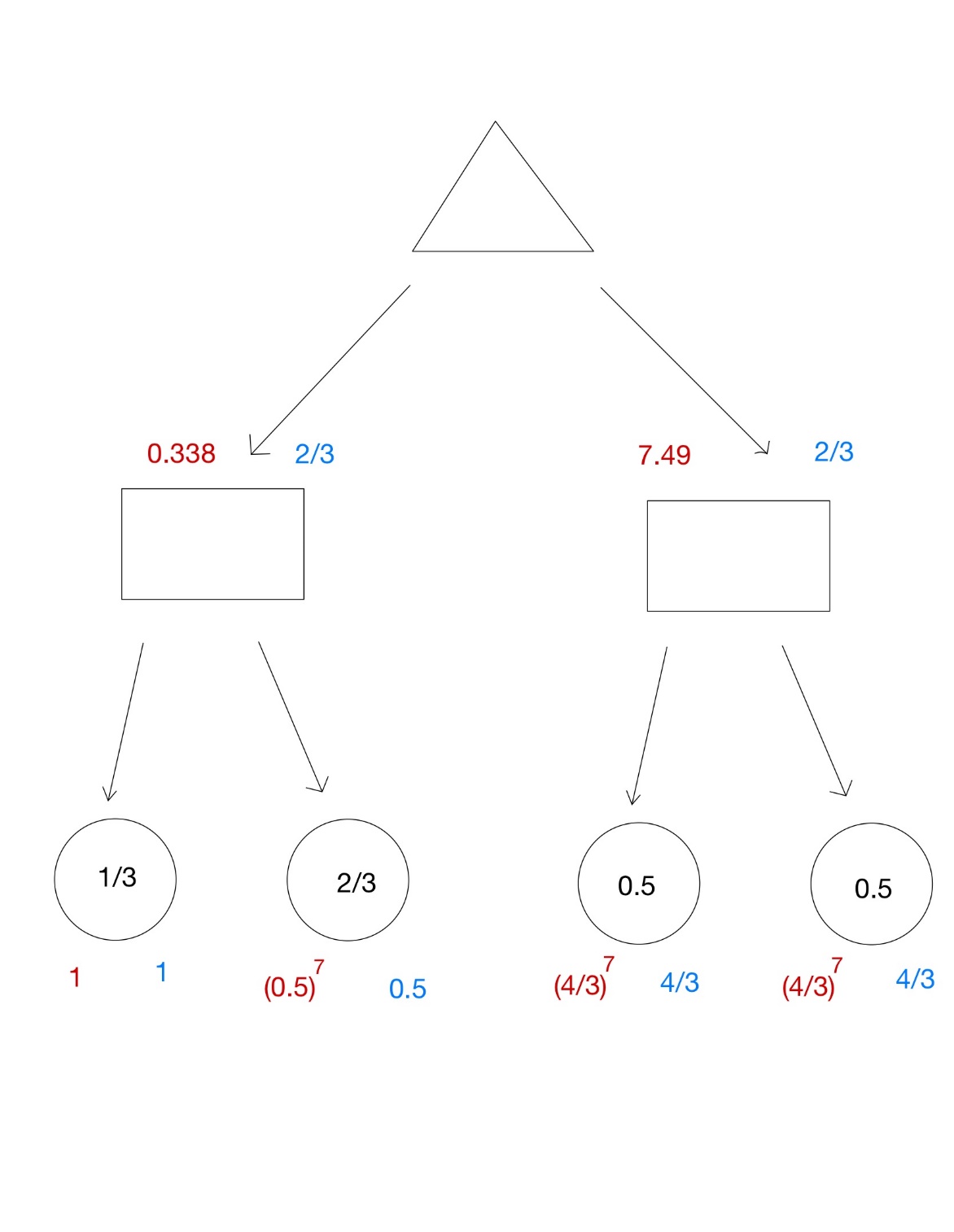
If the time limit is long enough for both agents to search the whole tree, then the chosen step will be the same, but if both are interrupted at the same time, then it is possible that the agents may choose different steps. This is because of the difference in runtime – the alpha-beta agent may reach deeper levels than the minimax agent resulting in a different choice of the step.

**Part 4**

1. It is a bug. A-B pruning in general returns the “best of the worst” – as in it assumes that the enemy is optimal and calculates the best value for the agent given the latter. Expectimax on the other hand allows for enemy “mistakes” – as in given probabilities of enemy actions, it returns the best value of the expectations of the game roll-outs. Example of different values – setting: agent and enemy have one step left, agent starts first:

**Last part – open question**

* 1. Utility(state) =
  2. Yes, since it is possible that there undefined utilities which causes the system to crash.
  3. No, since the transformation preserves the order in the utilities space, therefore values may change but actions will not. The transformation’s effect is below.
  4. a. Yes, since it is possible that there undefined utilities which causes the system to crash.

b. Yes, below is an example of an expectimax tree that will yield a different action. The triangle represents a max node, while the rectangles are chance nodes. Blue numbers are before the change and red ones are after. Before the change the max node can choose either of its children, but after the change it will choose the right child.

* 1. 1. Yes she can. For every state, given the function , .  
      Since α represents the lower bound of the maximal minimax value of a max vertex and also represents a lower bound of , updating α to be max(𝛼, 𝑓(𝑠) − 2) will make it a tighter lower bound which may induce more pruning. Analogously, β represents the upper bound of the minimal minimax value of a min vertex and also represents a upper bound of , updating β to be min(β, 𝑓(𝑠) + 2) will make it a tighter upper bound which may also induce more pruning.

If (𝛼 ≥ 𝑓(𝑠) + 2) then it is guaranteed that therefore there is no need calculate the subtree and it can be pruned, also if (𝛽 ≤ 𝑓(𝑠) − 2) then and the subtree can be pruned.

This way the number of visited vertices can be decreased, while maintaining correctness of the algorithm.

The pseudo-code:

AlphaBeta(State, Agent, Alpha, Beta):

If G(State) then return h(State, Agent)

**𝛼 = max(𝛼, 𝑓(𝑠) − 2)**

**𝛽 = min(𝛽, 𝑓(𝑠) + 2)**

**if (𝛽 ≤ 𝑓(𝑠) − 2) 𝑡ℎ𝑒𝑛 𝑟𝑒𝑡𝑢𝑟𝑛 𝑓(𝑠)− 2**

**if (𝛼 ≥ 𝑓(𝑠) + 2) 𝑡ℎ𝑒𝑛 𝑟𝑒𝑡𝑢𝑟𝑛 𝑓(𝑠) + 2**

Turn ← Turn(State)

Children ← Succ(State, Agent)

If Turn = Agent then

CurMax ← -∞

Loop for c in Children

v ← RB-AlphaBeta(c, Agent, Alpha, Beta)

CurMax ← Max(v, CurMax)

Alpha ← Max(CurMax, Alpha)

If CurMax ≥ Beta then return ∞

Return(CurMax)

else

CurMin ← ∞

Loop for c in Children

v ← RB-AlphaBeta(c, Agent, Alpha, Beta)

CurMin ← Min(v, CurMin)

Beta ← Min(CurMin, Beta)

If CurMin ≤ Alpha then return -∞

Return(CurMin)

1. No , the f function cannot be used on a resource bound alpha-beta algorithm, since there is no correlation between the f function and the heuristic function.

If the f function was used and a V value was returned it is not guaranteed that there is a strategy to reach the vertex with a heuristic value of at least V in D steps (D is the depth limit).

An example of that is running rb alpha-beta with depth 1 on the graph below, with the red numbers being the f values and the blue numbers being the heuristic values.

