Anton Hung
251023489
Ahung44

1.
The turing test is designed to see if it can determine, between two subjects, which is the human and which is the computer. Both subjects are trying to convince the tester that they can think and behave like a human. If the tester cannot tell a difference, then it is said that the AI system passed the turing test.

It is not a good evaluation of the AI we are studying. Our programs are far too simple to fool anyone into thinking it is human. Our programs are simple automation and designed to solve intelligent, logical problems and games but do not expand into the wide scope of all things involved in humanity such as conversation, innovation, etc.

To measure creativity, similarly to the turing test, there needs to be a human adjudicator. A random thing that already exists is chosen (movie, appliance, music, etc.) The human and the AI are both tasked with creating something of that nature. If the adjudicator cannot determine which was made by a human, which was made by the AI, then the AI passes the test

2.
The time complexity for depth first is exponential ($O(b^m)$). Space complexity is linear.
Time complexity for breadth first is also exponential ($O(b^s)$) where $s \leq m$. Space complexity is exponential
Iterative deepening search is a blend of depth-first search and breadth-first search. Unlike depth-first, it has a depth limit, and then backtracks to the highest unexplored level. This means we have the avantage of using dfs' fringe linear space complexity, while also staying shallow in the search tree. The point is that it is depth first search, but returns to the top after it reaches its depth limit. Space complexity is always linear, and we don't need to go all the way to the root nodes before returning to shallower nodes.
Also unlike depth-first, iterative deepening is complete. It will find the solution if it is present.
Iterative deepening is optimal in terms of finding the least-cost solution if every action is the same cost.

3. Good heuristic functions are non-trivial. If the goal is reached, the heuristic value should be 0. They are admissible. The heuristic does not return an estimated cost greater than the actual cost. They are consistent. The difference in estimated cost after a single action will not exceed the cost of the action. Therefore there are not sudden large changes in heuristic values when the action cost is small. The heuristic also does not get larger as we get closer to the goal.
To create these heuristics we can ignore constraints which would ensure that the heuristic is admissible. We can also use a semilattice to take the maximum of two admissible heuristics. This way the heuristic is closer to the actual cost while remaining admissible.

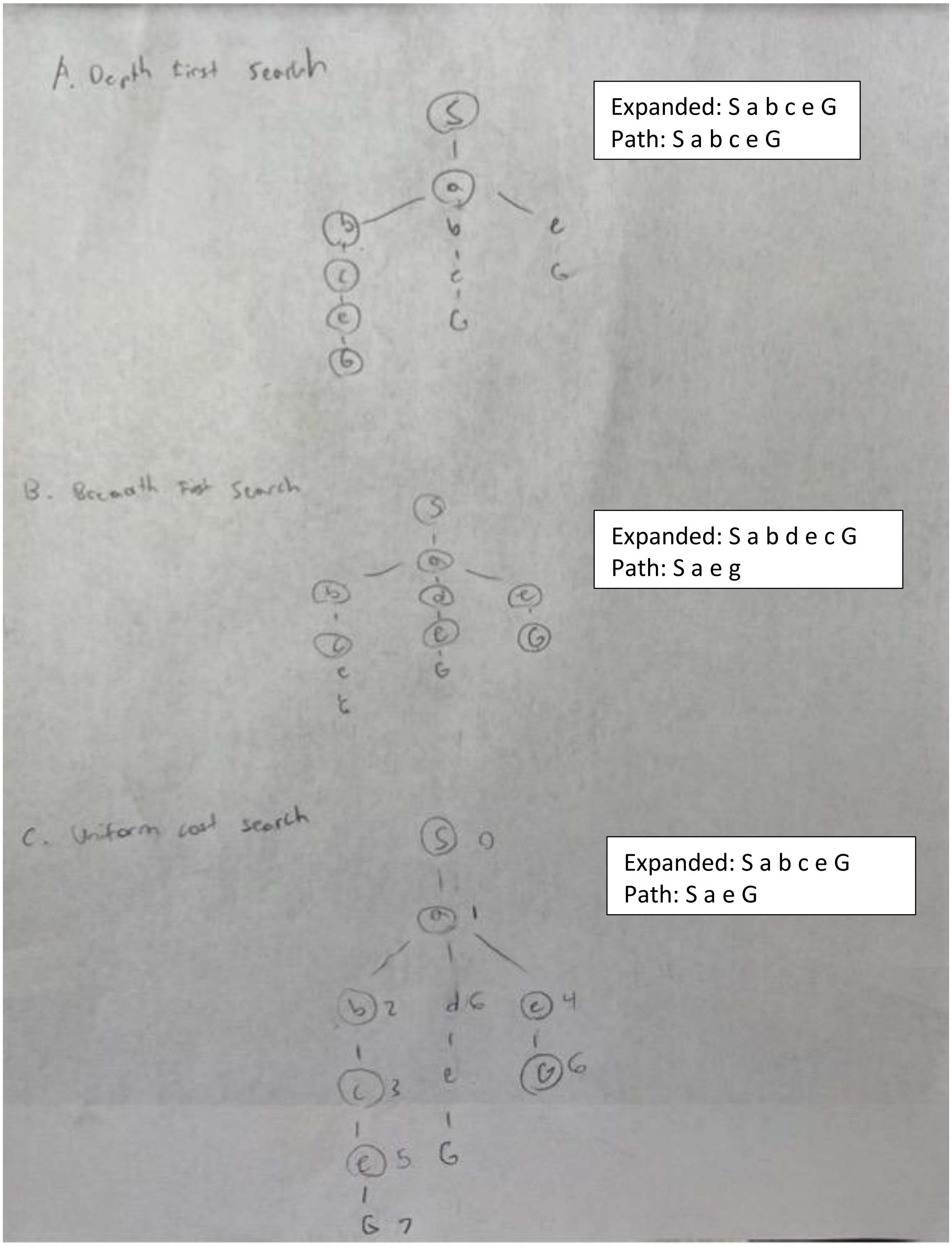
4. True/False

A. TRUE
   UCS opens up the least cost nodes first. So, when it finds the solution, it is guaranteed to be the least cost path to the solution. Whether it is tree search or graph search does not make a difference.

B. TRUE
   Iterative deepening takes the advantage of a linear fringe complexity from dfs. But it has a depth limit.

C. TRUE
   If two heuristics are admissible, then neither of them ever overestimate the cost. Therefore their average cannot possibly overestimate either. The average heuristic lies in between the maximum and minimum of the two heuristics. Since the maximum and minimum are both admissible and consistent, then anything in between must be admissible and consistent as well.

5.

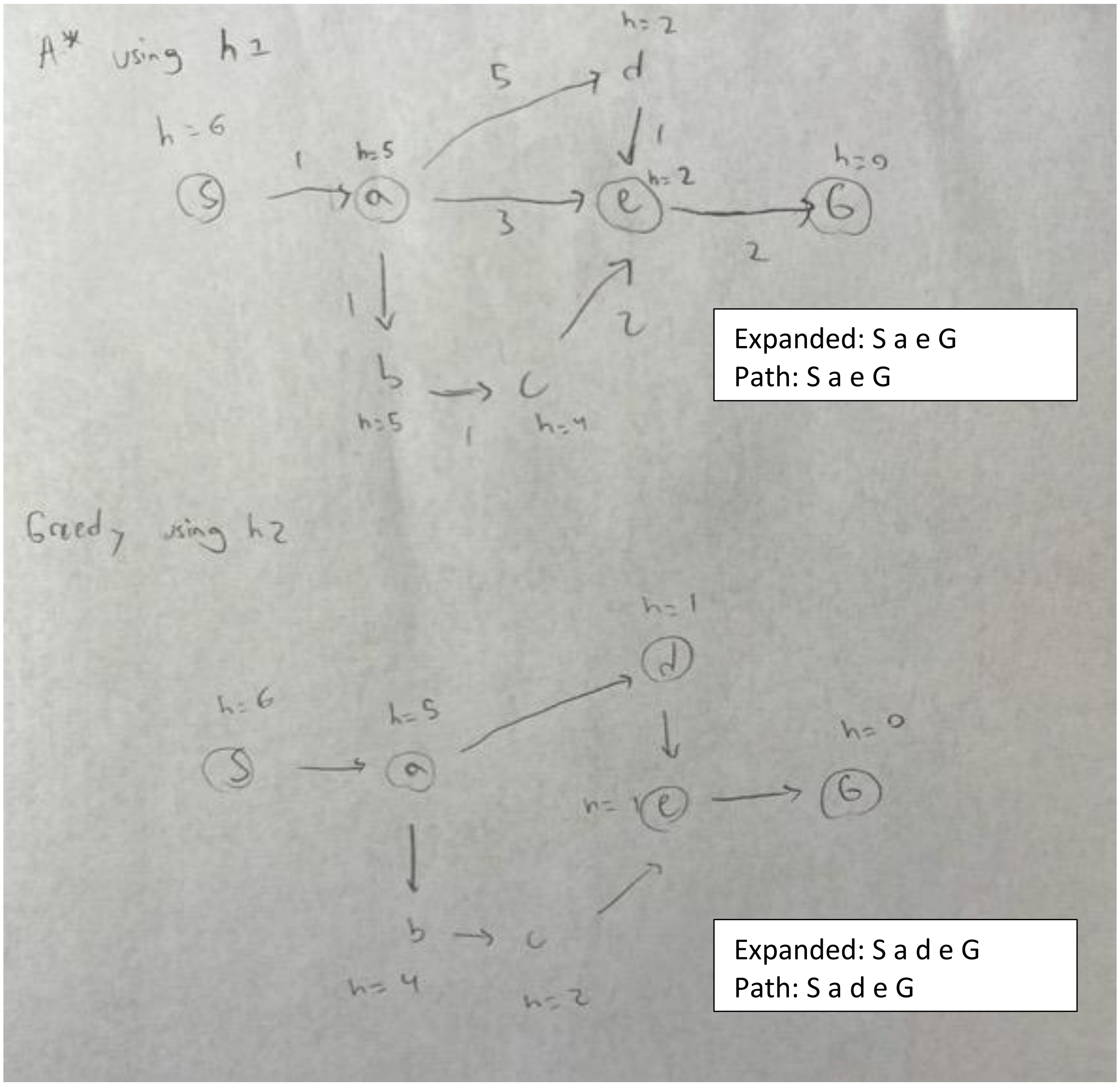


Expanded: S a b c e G
Path: S a b c e G

Expanded: S a b d e c G
Path: S a e g

Expanded: S a b c e G
Path: S a e G

5. D.
Admissible: h1, h2
Consistent: h1



6.

A. The state space is a set representing the distribution of the 20 ounces of water across the three jugs.

A state is represented by 3 tuples representing the 3 jugs. Each tuple contains 2 values; the first is the capacity of the bottle (19, 13, or 7), the second is the amount of water in the jug (0-19, 0-13, 0-7) and cannot be greater than the value of the first number. The first number stays the same while the second can change. The sum of the second numbers in each tuple must remain constant.

B. Initial state {(19,0), (13,13), (7,7))},
Goal state {(19,10), (13,10), (7,0)}

C. An action consists of pouring the water from one jug to another. This means one jug decreases by x ounces and another increases by exactly x ounces. You cannot have only a decrease or only an increase, both must come together. The amount of water in a jug cannot increase if it is full (when capacity of jug = amount of water in the jug). The cost of an action is equal to how many ounces are transferred. An action is represented by a tuple of two numbers (x,y,z); x represents the capacity of the jug that water is being removed from, y represents the capacity of the jug that water is being added to, z represents the amount of ounces transfered. (19, 13, 10) means 10 ounces are being transferred from the 19-ounce jug to the 13-ounce jug.

A successor is a triple: (state, action, cost). The successor function returns all the valid successors. A successor is valid if every tuple (x,y) in the state has a x≥y. If pouring the water from one jug into a second would overflow the second jug, then the second jug is filled just to its capacity, and the first will have some leftover.

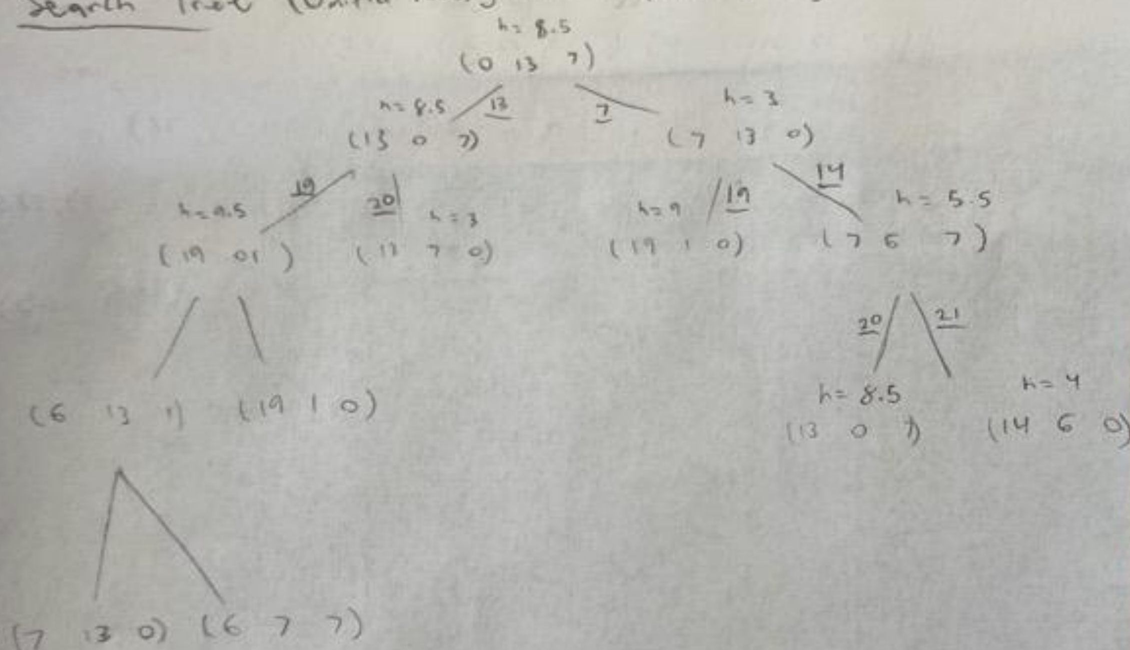
D. IN PICTURE ON NEXT PAGE

E. A heuristic could measure how many ounces away each jug is from the goal, and taking the average of the largest 2. The average volume change needed is always going to be less than the volume change needed for the jug that is farthest from the goal state. It will also be less than the actual amount of actions needed and this makes it admissible. When the goal is reached, the average distance from the goal will be 0, so the heuristic is admissible

6. D and part of F

6. (x y z) where x: volume of 19-ounce jug
y = volume of 13-ounce jug
z = volume of 7-ounce jug

Search Tree (Omitted reversing back to parent states)



D

DFS

Fringe: (x y z)

(0 13 7)

(13 0 7) (7 13 0)

(19 0 1) (13 7 0) (7 13 0)

(6 13 1) (19 1 0) (13 7 0) (7 13 0)

(7 13 0) (6 7 7) (19 1 0) (13 7 0) (7 13 0)

6. F. continued

F

A*

Fringe : ((x y z), h + path cost)

((0 13 7), 8.5)

((7 13 0), 10) ((13 0 7), 21.5)

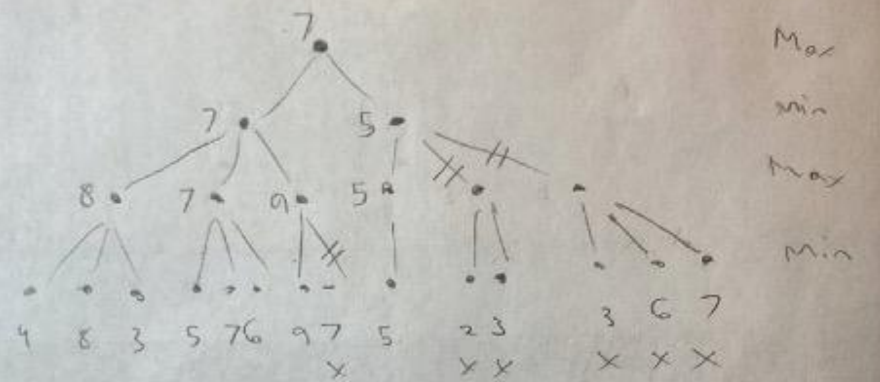
((7 6 7), 19.5) ((13 0 7), 21.5) ((19 1 0), 28)

((13 0 7), 21.5) ((14 6 0), 25) ((19 1 0), 28) ((13 0 7), 28.5)

((13 7 0), 23), , , , ((19 0 1), 28.5)

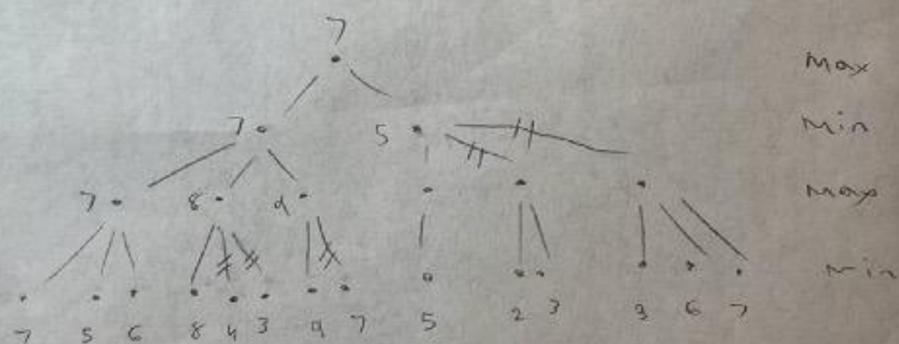
7.

7. A.

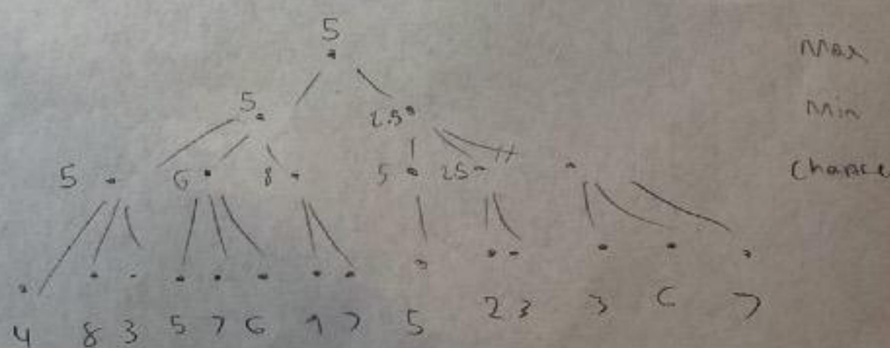


B. Max takes 7

C.



D.



8.

A.

The variable for every class is which class is being taught: (1, 2, 3, 4 or 5)
The domains are which professor is teaching the class: (A, B or C)
Constraints:

> Classes occurring at the same time must be taught by different professors.
> Professors may only teach a class that they are available for.

B, c, d: