Henry Jacobs

CS 380 – Artificial Intelligence

Jeff Popyack

May 12, 2022

**HW3 – hmj32**

**Question 1A – Algorithm A\* (8-Puzzle)**

H0(node): 0

* If our heuristic is always zero, then it will never be overestimating the distance to the goal and would be seen as admissible.

H1(node) : Number of Tiles out of Place

* This heuristic is admissible because since one is unable to move more than tile at a time, so it’s going to take you at least the same amount of moves to reach the goal as there are tiles out of place.

H2(node): Sum of distances out of place

* The sum of the out of place distances will never overestimate the actual distance to the goal since a tile will have to make at least the same amount of moves in order to get to its goal position. Therefore, this is an admissible heuristic.

H3(node): 2\*DT(node)

* H3 is an admissible heuristic because it won’t overestimate the distance to the goal since we cannot solve this puzzle with our current goal. This means that we’ll never overestimate the distance.

H4(node): h2(node) + 3\*S(node)

* This is not an admissible heuristic as it will over-estimate the distance to the goal. For example, the puzzle below can be solved in 5 moves, but this heuristic estimates it will take 9 moves.

2 8 3

1 6 4

7 0 5

H5(node): h1(node) + h3(node)

* If there are no tile reversals on the board (such as on the 8-puzzle above), then h3(node) = 0. In this case, we are left with only the number of tiles out of place, which we know is an admissible heuristic. In the case of there being 2 direct reversals on the board, then we know that h3(node) = 4 and h1(node) = 4, so this means that h5(node) = 8. Since the puzzle with two direct tile reversals cannot be solved using our current goal state, then we know this won’t overestimate the distance to the goal.
* As a result, H5(node) is an admissible heuristic.

H6(node): h2(node) + h3(node)

* Again, since h2(node) is already an admissible heuristic and we know that direct tile reversal puzzles cannot be solved given our current goal state, then this heuristic will not overestimate the distance to the goal.
* If no direct tile reversals, then our heuristic just becomes h2(node) which is admissible and if there are direct tile reversals, the puzzle cannot be solved and our heuristic will not overestimate the distance to the goal (since we can keep going forever and not reach it!).

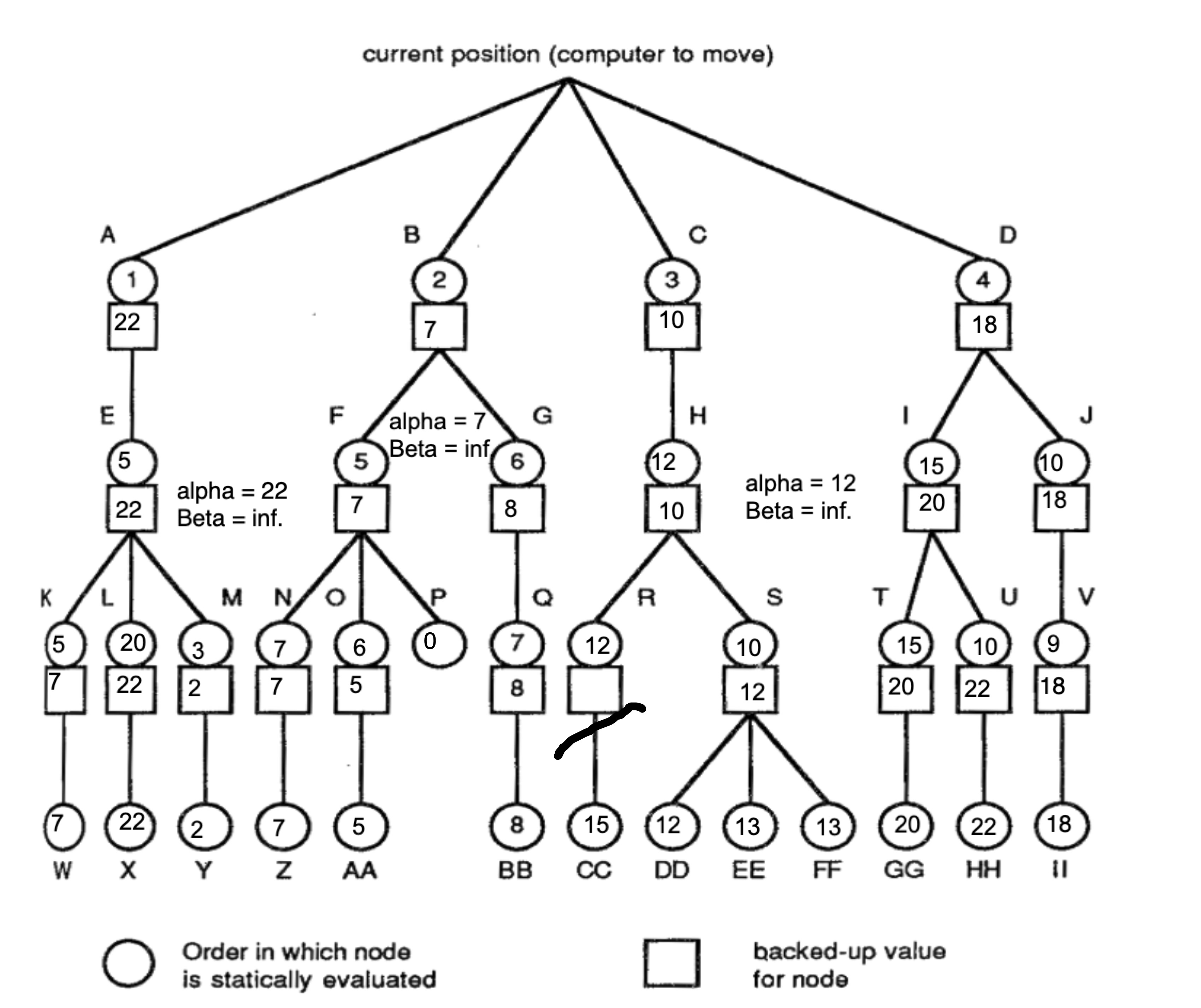
H7(node) -- Maximum of all Admissible Heuristics

Since H7 is always picking the maximum of the admissible heuristics, it will always be the largest and it will always be admissible.

Precedence from Least to Greatest

H0 ≤ H3 ≤ H1 ≤ H5 ≤ H2 ≤ H6 ≤ H7

**Question 1B.**



To be completely honest, I struggled with this question. I know it did it incorrectly, I just had trouble understanding the static evaluation part of the problem. I felt like I hadn’t seen a question like this before, so it really kind of stumped me.

1C. Given that it takes the computer 1/7 seconds to move, I believe it would choose Node B, given that it has a higher static evaluation function, and it hasn’t been able to explore the other parts of the graph quite yet.

1D. Given that the computer would now take 1/8 seconds to make a move, I believe it would still pick Node B since it has explored node B in depth and found that it has a higher real value than the static evaluation found at location A.