Phot Koseekrainiramon z5387411 COMP3411

Assignment 2 – Heuristics and Search

**Question 1: Search Strategies for the 15-Puzzle**

(a)

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Start State | BFS | | IDS | | Greedy | | A\* | |
| start1 | 12 | 10978 | 12 | 25121 | 12 | 59182 | 12 | 30 |
| start2 | 17 | 344890 | 17 | 349380 | 17 | 19 | 17 | 35 |
| start3 | 18 | 641252 | 18 | 1209934 | 22 | 59196 | 18 | 133 |

(b) Breadth-First Search seems to be the second worst in terms of node expansion efficiency, slightly better than IDS. The lengths of the resulting path are similar to all other strategies, except for start3, where Greedy produced the most lengths.

Iterative Deepening Search seems to have worse node expansion efficiency than all other strategies, with the highest number of nodes expanded in all starts. While IDS finds paths of similar lengths to BFS and A\*, it expands far more nodes to achieve it.

Greedy expands a larger number of nodes for start1 compared to BFS and IDS; however, it expands significantly fewer nodes for start2 and start3 compared to BFS and IDS. At the same time, it expands the least number of nodes for start2 among all strategies. The number of nodes expanded seems to be about the same for start1 and start3. On average, the node expansion efficiency is better than BFS and IDS. However, it produces a longer path in start3 compared to the other.

A\* expands the least number of nodes in all starts except start2, which Greedy beats. The length of the resulting paths is the same as the other, Making A\* the most efficient overall.

**Question 2: Heuristic Path Search for 15-Puzzle**

(a) Let and we know that . Then, and . This shows .

Then, try fitting into we will get which is now the same as . So, we can say that minimizing is the same as .

Hence is also admissible, so the Heuristic Path is optimal.

(b)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | start4 | | start5 | | start6 | |
| IDA\* Search | 45 | 545120 | 50 | 4178819 | 56 | 169367641 |
| HPS, | 47 | 523052 | 54 | 857155 | 58 | 13770561 |
| HPS, | 47 | 29761 | 56 | 64522 | 60 | 265672 |
| HPS, | 55 | 968 | 62 | 5781 | 68 | 9066 |
| HPS, | 65 | 9876 | 70 | 561430 | 80 | 37869 |

(c) The path length increases as the value of rises from 1.0 to 1.4. However, for start4, the path length is the same when the values of are 1.1 and 1.2.

The number of expanded nodes decreases as the value of increases from 1.0 to 1.3. However, when the value of is 1.4, the number of expanded nodes increases compared to when the value of w is 1.3. Some even have more nodes than when the value of is 1.2 (start5).

**Question 3: Graph Paper Grand Prix**

(a)

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Optimal Sequence of Actions | | | | | | | | | |  |  |
| 1 | + | - |  |  |  |  |  |  |  |  | 2 | 1 |
| 2 | + | 0 | - |  |  |  |  |  |  |  | 3 | 1 |
| 3 | + | 0 | 0 | - |  |  |  |  |  |  | 4 | 1 |
| 4 | + | + | - | - |  |  |  |  |  |  | 4 | 2 |
| 5 | + | + | - | 0 | - |  |  |  |  |  | 5 | 2 |
| 6 | + | + | 0 | - | - |  |  |  |  |  | 5 | 2 |
| 7 | + | + | 0 | - | 0 | - |  |  |  |  | 6 | 2 |
| 8 | + | + | 0 | 0 | - | - |  |  |  |  | 6 | 2 |
| 9 | + | + | + | - | - | - |  |  |  |  | 6 | 3 |
| 10 | + | + | + | - | - | 0 | - |  |  |  | 7 | 3 |
| 11 | + | + | + | - | 0 | - | - |  |  |  | 7 | 3 |
| 12 | + | + | + | 0 | - | - | - |  |  |  | 7 | 3 |
| 13 | + | + | + | 0 | - | - | 0 | - |  |  | 8 | 3 |
| 14 | + | + | + | 0 | - | 0 | - | - |  |  | 8 | 3 |
| 15 | + | + | + | 0 | 0 | - | - | - |  |  | 8 | 3 |
| 16 | + | + | + | + | - | - | - | - |  |  | 8 | 4 |
| 17 | + | + | + | + | - | - | - | 0 | - |  | 9 | 4 |
| 18 | + | + | + | + | - | - | 0 | - | - |  | 9 | 4 |
| 19 | + | + | + | + | - | 0 | - | - | - |  | 9 | 4 |
| 20 | + | + | + | + | 0 | - | - | - | - |  | 9 | 4 |
| 21 | + | + | + | + | 0 | - | - | - | 0 | - | 10 | 4 |

(b) Note: The following is the number of consecutive ‘-‘s at the end of the sequence, so for all cases. The following is the number of ‘+’s at the beginning of the sequence.

Given identity:

From the table above. When and then respectively. You will notice that the pattern falls , which is valid for the first case of the given identity where .

When and then respectively. You will notice that the pattern falls , which is valid for the second case of the given identity where .

When and then respectively. You will notice that the pattern falls , which is valid for the third case in the given identity where .

Thus, is valid.

(c) For the base case where , from (b):

equals to since .

Where , the number of sequences () would be decreased by since starting off at the velocity of means we don’t need to spend steps to accelerate the velocity to from a velocity of 0. Thus, needing fewer steps compared to when .

With is the distances required to decelerate from a velocity of to 0 . It would take distances to accelerate to the velocity of from a velocity of 0 . This is why we need to sum with if we start at the velocity of and replace the original from with the value (from ).

This proves .

(d) Since is the distances required to decelerate from a velocity of to 0. Then, it is impossible to reach when since the agent would surpass the goal before stopping. So, the agent must accelerate again but in the opposite direction to stop at the goal. It has the same logic as the previous question, but in the opposite, it would take steps to decelerate from a velocity of to a velocity of 0. This means we need to add an extra to the number of sequences instead.

After reaching the point where , the agent needs to start accelerating in the opposite direction. With the total distances of from the goal where the goal is at .

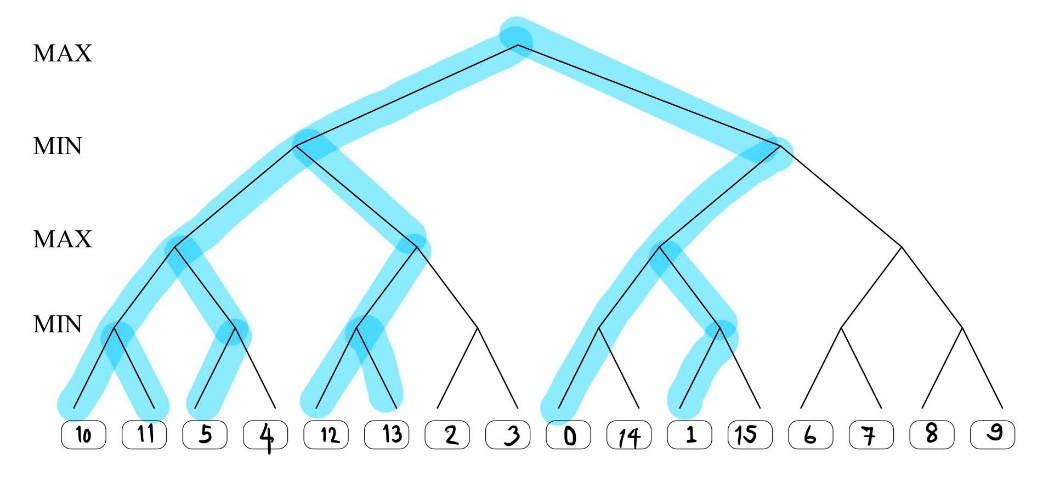
Thus, .

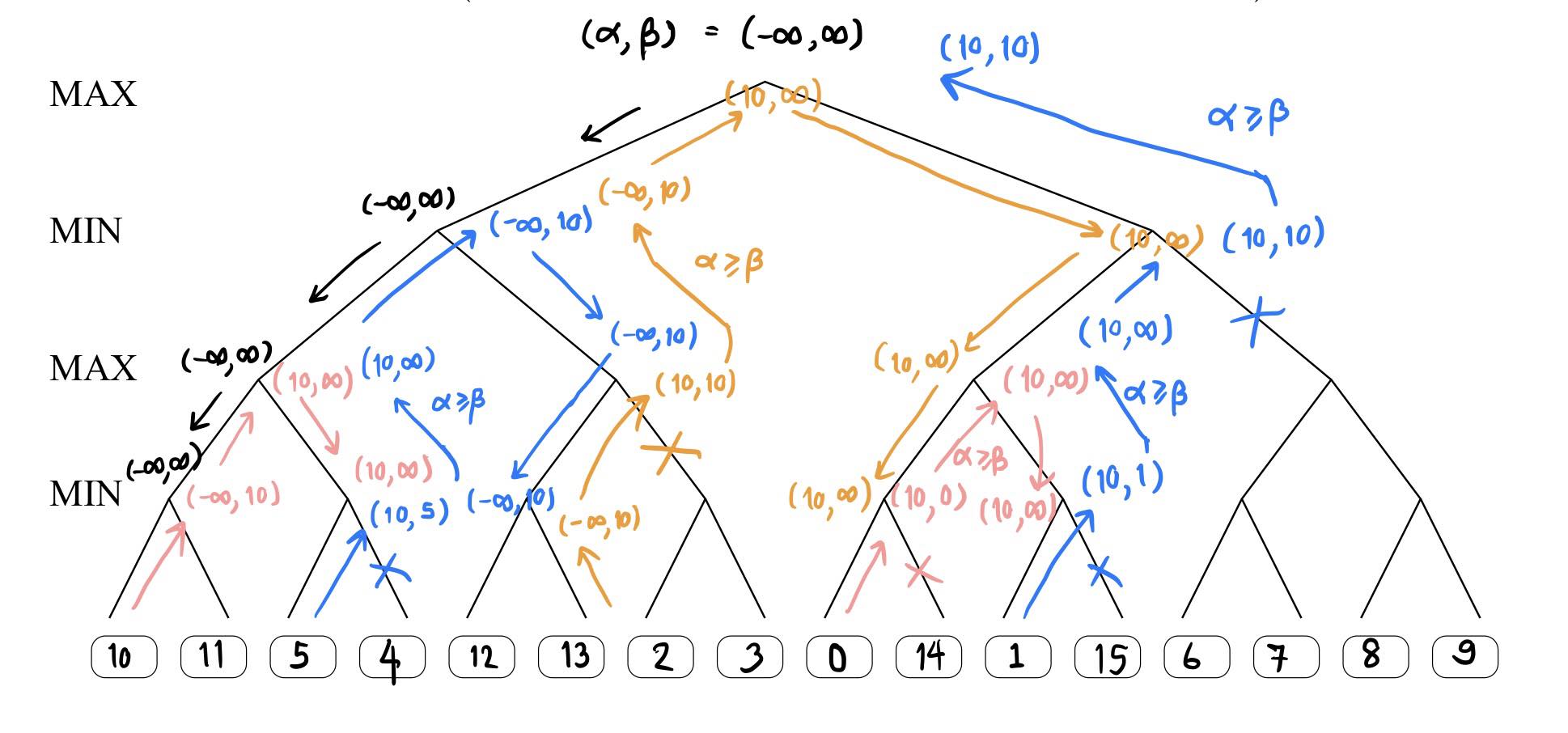
(e) Where the start position of the agent is and the position of the goal be . Then, the total distances the agent travels horizontally (x-axis) is going to be , and vertically (y-axis).

With , however, we also need to take velocities into account since they can be both positive and negative, as in the previous example that the agent needs to reverse its direction. With the initial velocities are and for horizontal and vertical, respectively, we can derive the admissible heuristic as follow

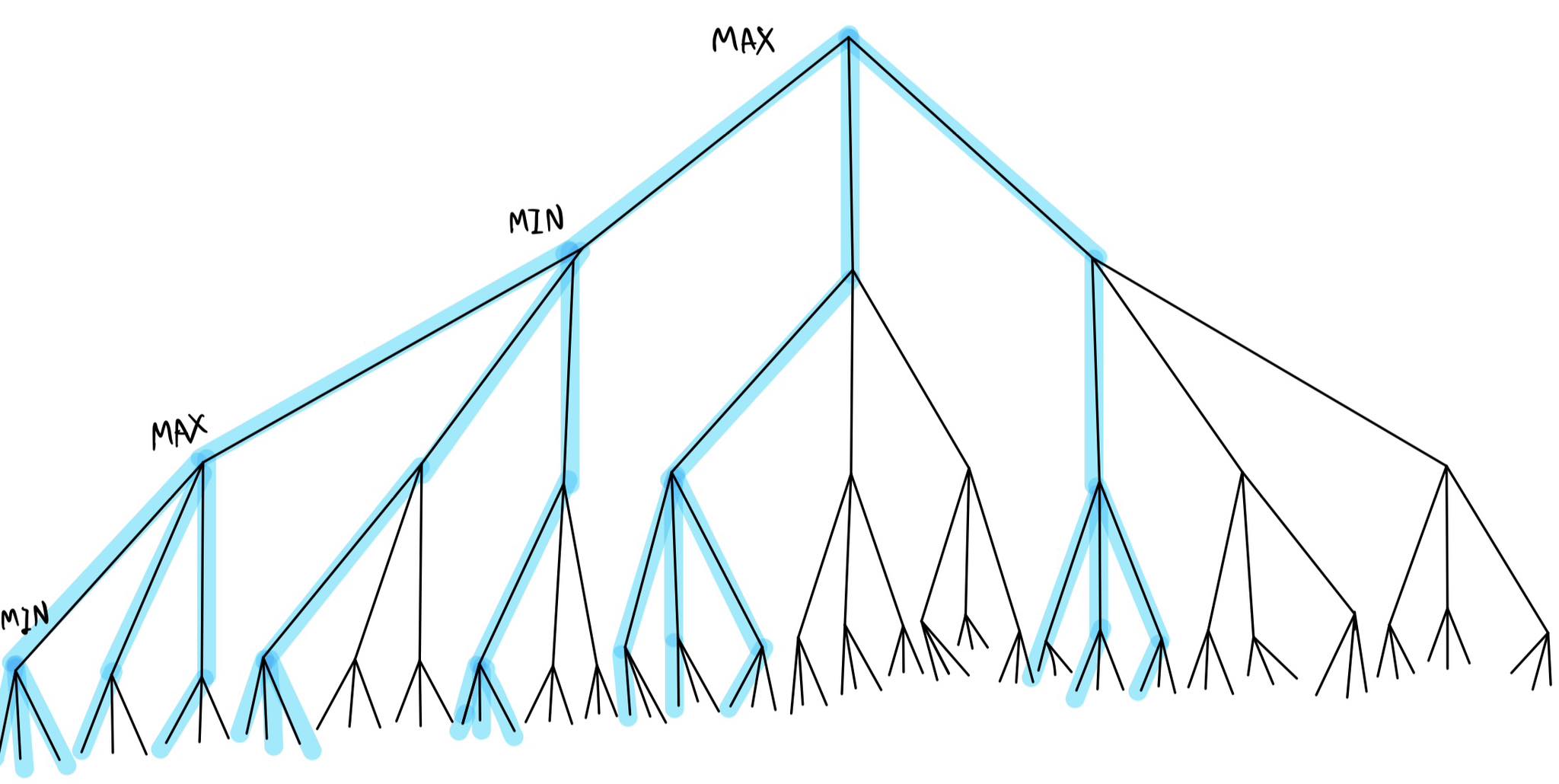
**Question 4: Game Trees and Pruning**

(a, b)





(c) 17 out of 81 leaves will be evaluated.



(d) The time complexity of alpha-beta search, if the best move is always examined first, would be when is the number of children in an internal node and is the depth of the tree. Since the best move always occurs at the left side of the tree, ignoring the rest when possible, would reduce the number of nodes needed to explore to approximately the square root of the original number of children per node.