Assignment 2 – Heuristics and Search

Question 1: Search Strategies for the 15-Puzzle:

(a):

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Start State | BFS | IDS | Greedy | A\* |
| Start1 | Expanded: 10978.  Length: 12. | Expanded: 25121.  Length: 12. | Expanded: 59182.  Length: 12. | Expanded: 30. Length: 12. |
| Start2 | Expanded: 344890.  Length: 17. | Expanded: 349380.  Length: 17. | Expanded: 19. Length: 17. | Expanded: 35. Length: 17. |
| Start3 | Expanded: 641252.  Length: 18. | Expanded: 1209934. Length: 18. | Expanded: 59196.  Length: 22. | Expanded: 133. Length: 18. |

(b):   
BFS:

BFS guarantees the shortest path and can be seen in all instances to have achieved the goal with optimal length. However, it tends to expand a significant number of nodes, especially as the complexity of the start state increases. Of course, it is the way how BFS arithmetic works. It is evident from a mass of number of nodes it expanded for Start2 and Start3, indicating a high memory requirement and potentially longer run times in practice.

IDS:

Like BFS, IDS also guarantees the shortest path, which is evident from the lengths being optimal across all starts. However, it expands a vast number of nodes, even more than BFS in all cases here, which means it will achieve a best answer may be different with BFS but with more cost.

Greedy Search:

Greedy Search uses the Manhattan Distance heuristic to direct the search process, resulting in a significant reduction in the number of expanded nodes for Start2 and Start3. However, it may not be able to guarantee the shortest path, such as in Start3, where the path length is 22 but the optimal length is 18. In general, use greedy search is a risk choice, it just has some possibility to reduce to the number of nodes expanded with a shortest path.

A\* search:

A\* Search is more balance than BFS, IDS or Greedy Search when finding the shortest path while also being more efficient in terms of nodes expanded, because it considers both the cost to reach the current node and the heuristic estimate. The table above shows that A\* expanded far fewer nodes than BFS and IDS for all starts while maintaining the optimal path length, which proves its efficacy in both node expansion and path optimality.

Question 2: Heuristic Path Search for 15-Puzzle

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Start4 | | Start5 | | Start6 | |
| IDA\* SEARCH | 45 | 545120 | 50 | 4178819 | 56 | 169367641 |
| HPS, w = 1.1 | 47 | 523052 | 54 | 857155 | 58 | 13770561 |
| HPS, w = 1.2 | 47 | 29761 | 56 | 64522 | 60 | 265672 |
| HPS, w = 1.3 | 55 | 968 | 62 | 5781 | 68 | 9066 |
| HPS, w = 1.4 | 65 | 9876 | 70 | 561430 | 80 | 37869 |

However, when w=1.4, the number of expanded nodes increases, which might be due to excessive optimism heuristic. Overweighting may lead to heuristics emphasizing estimated costs while neglecting actual path costs, causing the search to deviate from more expensive paths, as this time the arithmetic does not care the cost too much.