AI-Assignment-1

Mohd.Sayemul Haque, Roll:53

27 February 2020

1 Introduction

N-puzzle is a well-known problem used for a long time by Artificial Intelligence (AI) researchers as a benchmark for comparing the performance of search algorithms. Given a square grid with only one empty and 8 puzzle pieces, the goal is to achieve a certain orientation of the puzzle pieces.The only restriction is you can only move the adjacent pieces to the empty slot.The goal is to implement various classical AI search algorithms for solving n-puzzle problem. It also entails performance analysis of these algorithms. The algorithms are: breadth first search (BFS), uniform cost search (UCS), depth limited search (DLS), iterative deepening depth first search (IDS), greedy best first search (GBFS), and A* search.

2 Result analysis

To analyze and compare between all the algorithms, they were run on the same data sets. Their performance like, time taken, node generated and steps taken were observed and plotted with respect to the optimal solution steps.

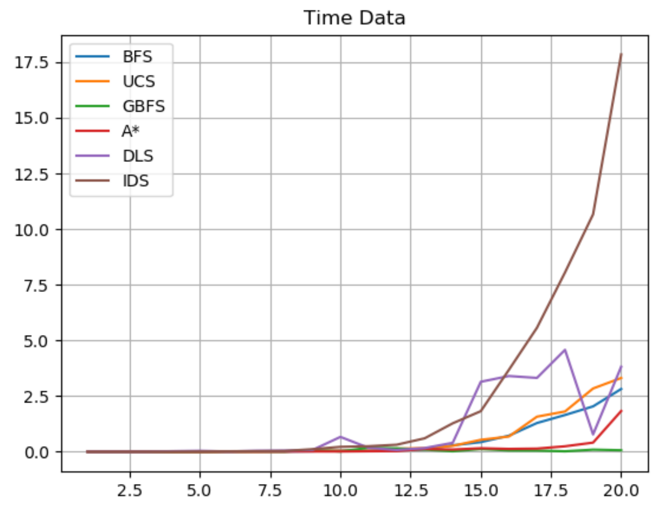

Figure 1: Time analysis

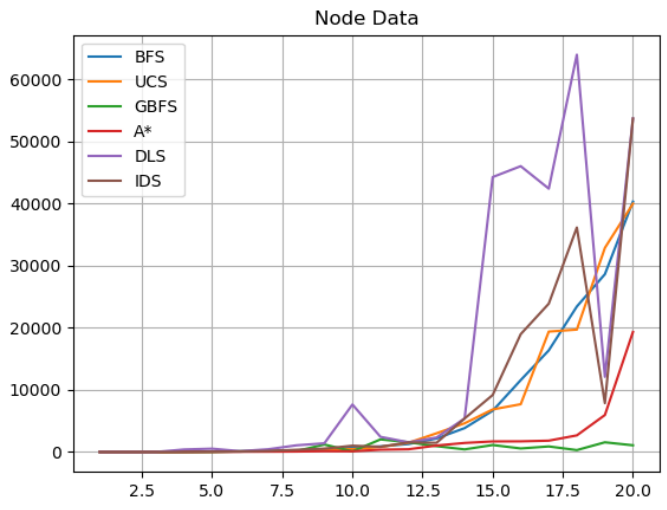

Figure 2: Nodes generated

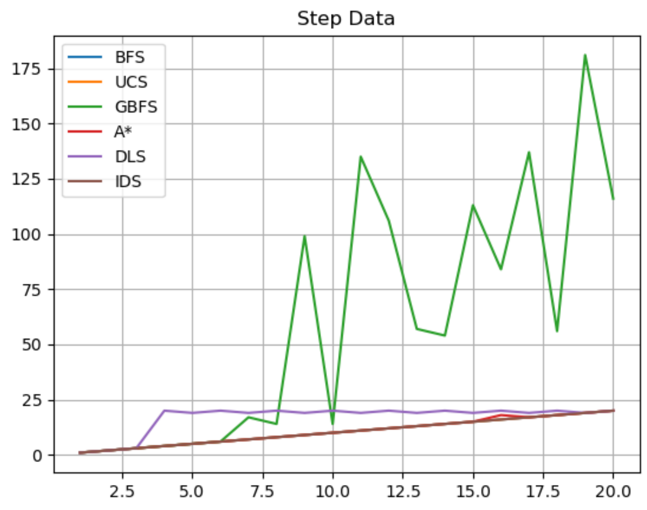

Figure 3: Steps taken

From the time analysis graph we can conclude that GBFS and A* takes the list amount of time where as IDS takes the most due to iterative approach it takes.Whereas, BFS and UCS are quite similar in this particular case(as each step cost is equal to one). DLS is also a computation heavy approach right after IDS.

The node analysis also gives a similar outlook of all the algorithms

On the other hand, steps analysis graph shows that, except GBFS and DLS, othe algorithms always give the optimal solution. Although both GBFS and A* use heuristic functions, A* chooses the best possible action and proceeds in that path as it doesn't generate redundant child states, as the algorithm will expand the node with the least previous state score plus the next state score. Whereas GBFS only checks the next state score. Here we can also say that heuristic functions improves the computational cost by a lot.

3 Challenges faced

The first challenge was to understand all the algorithms. The most difficult part was implementing them, using the pseudo-code of the book and with some online help. Another challenge was to make the code as modular as possible. Some discussion with other students also assisted in completing this assignment