Report PA1
Manual For Running:
- Run 'make' command
- Run './p1'
- Run input

Data:
Nodes Visited (Time complexity)

Search Algorithm	Easy (1 3 4 8 6 2 7 0 5)	Medium (2 8 1 0 4 3 7 6 5)	Hard (5 6 7 4 0 8 3 2 1)
DFS	N/A	N/A	N/A
BFS	43	344	181365
IDS	63	798	4419719
Greedy (h1)	6	44	611
Greedy (h2)	6	44	125
A* (h1)	7	30	111149
A* (h2)	6	18	1277

**h1 is the number of tiles out of place and h2 is the sum of manhattan distance

Max Node List Size (Space complexity)

Search Algorithm	Easy (1 3 4 8 6 2 7 0 5)	Medium (2 8 1 0 4 3 7 6 5)	Hard (5 6 7 4 0 8 3 2 1)
DFS	N/A	N/A	N/A
BFS	36	318	24983
IDS	5	9	30
Greedy (h1)	7	40	434
Greedy (h2)	7	40	109
A* (h1)	8	28	30422
A* (h2)	7	17	977

**h1 is the number of tiles out of place and h2 is the sum of manhattan distance

Analysis

- The DFS method failed to reach the goal state on all easy, medium, and hard because it is essentially randomly searching for the goal. The search goes down one path or depth and has a very low chance of reaching the solution. The DFS time and space complexity could not be analyzed from the output because it was running indefinitely. However, the space complexity is $O(b * m)$ and time complexity is $O(b^m)$ in the worst case. The DFS uses less memory but is not a complete search. It is bad when there is deep or infinite depth state space.

- The BFS method, we see that the time complexity grows very quickly (exponentially) from the different tests. It's time complexity is $O(b^d)$ because it visits every node, and uses a lot of space. $O(b^{(d+1)})$. The advantage is that it is a complete search, but the disadvantage is the memory requirements.

- IDS advantage is that it is a complete search. It's the time complexity is $O(b^d)$ and uses little memory $O(b*m)$. We see this with the nodes visited and the max node list size. It is a factor slower than BFS method.

- The greedy search disadvantage is that it is not optimal, but it is very fast. It's time complexity is also exponential and time complexity is also exponential.

- A* advantage is that it is optimal, but it has a worse space complexity (exponential).

- Heuristics: the manhattan heuristic lead to better node selection in the priority queue than the tile out of place heuristic. We can see this by the number of nodes visited for h2 was much less than h1. The heuristic also take a very similar time to calculate the values, so h2 is a better choice for the informed search algorithms.