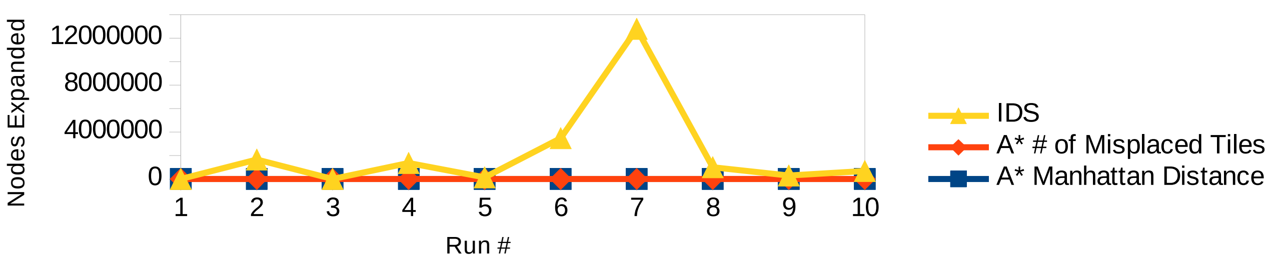
Alec Webb
CS580
HW1

1.) Eight Puzzle:

Results: Three Methods of Search, Values are # of Nodes Expanded.

Run:	A* Manhattan Distance	A* # of Misplaced Tiles	IDS
1	67	27	1251
2	90	277	1660779
3	88	72	4623
4	140	353	1368095
5	82	164	144406
6	89	385	3471866
7	319	939	12790603
8	230	416	1013670
9	104	180	298755
10	95	212	691253



Iterative deepening search (IDS) and A* search both have a run time complexity of O(b^d), however; because A* is an informed search with the use of a heuristic, and IDS is uninformed, the ability to find the solution more quickly is increased. The results of the 10 program executions align with this.

The implementation of A* used two different heuristics. The first was the Manhattan distance, and the second was the calculation of the number of misplaced tiles. Both heuristics are admissible in that they do not overestimate the actual solution. The Manhattan distance is admissible because every tile will move at least the number of positions required to reach the correct position which is <= to the correct solution. The number of tiles heuristic underestimates the cost of the actual solution even further numerically. The results show that the Manhattan distance heuristic dominates the number of misplaced tiles heuristic and therefore performs better overall, and that IDS is drastically worse overall.

The number of nodes expanded was measured by counting the number of nodes generated during the search until the final solution was found. In the chart above the two versions of A* performed similarly, while IDS greatly exceeded A* in regards to the number of expanded nodes.

As a final note, the performance of IDS was so slow in some cases that I was unable to come to a solution within a reasonable amount of time. As such, there are multiple logs which show a node expansion that was able to execute in under ~5 minutes.

2.) Two Jugs:

The condition that allows the two jug problem to be solved is that both capacities are relatively prime, or co-prime. The two integers, n and m representing the capacities must have a greatest common denominator of 1 in order reach solutions from 1...max(n,m). I have incorporated this into my solution by generating two random numbers in the range of 3 to 15 and pairing those that have a GCD equal to 1, adding these to a list until the list contains 10 pairs.

IDS performed well in finding the solutions to this problem compared to the Eight Puzzle problem, likely due to the set of possible solutions being greater, or that the number of expansions to reach a solution was smaller.