Building a Forward Planning Agent

Artificial Intelligence Nanodegree Ebrahim Jakoet

RESULTS OF PROBLEMS 1 & 2

The following problems and search algorithms were run for the first run.

PROBLEMS:

1. Air Cargo Problem 1
2. Air Cargo Problem 2

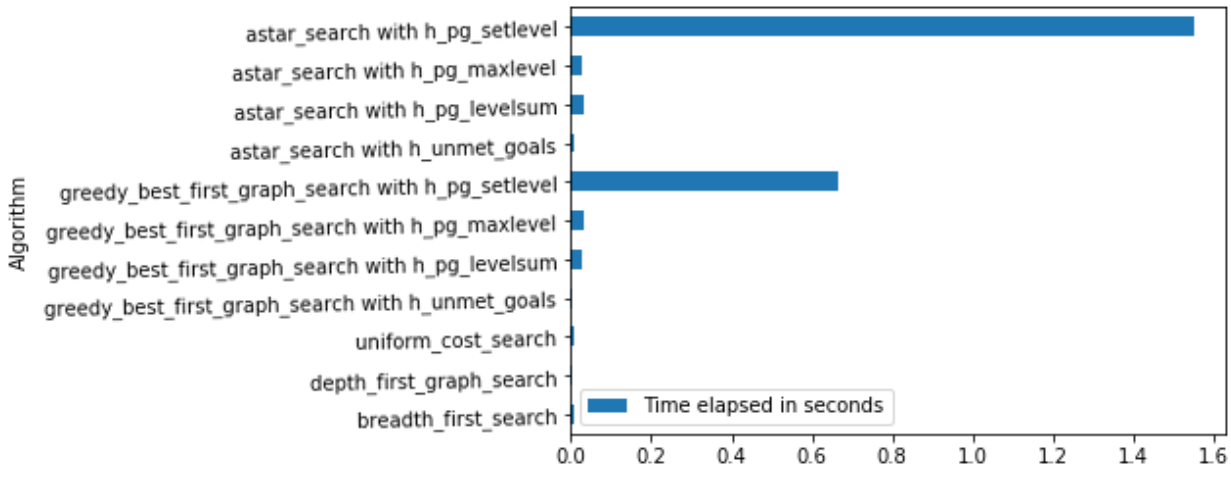
SEARCH ALGORITHMS:

1. breadth_first_search
2. depth_first_graph_search
3. uniform_cost_search
4. greedy_best_first_graph_search h_unmet_goals
5. greedy_best_first_graph_search h_pg_levelsum
6. greedy_best_first_graph_search h_pg_maxlevel
7. greedy_best_first_graph_search h_pg_setlevel
8. astar_search h_unmet_goals
9. astar_search h_pg_levelsum
10. astar_search h_pg_maxlevel
11. astar_search h_pg_setlevel

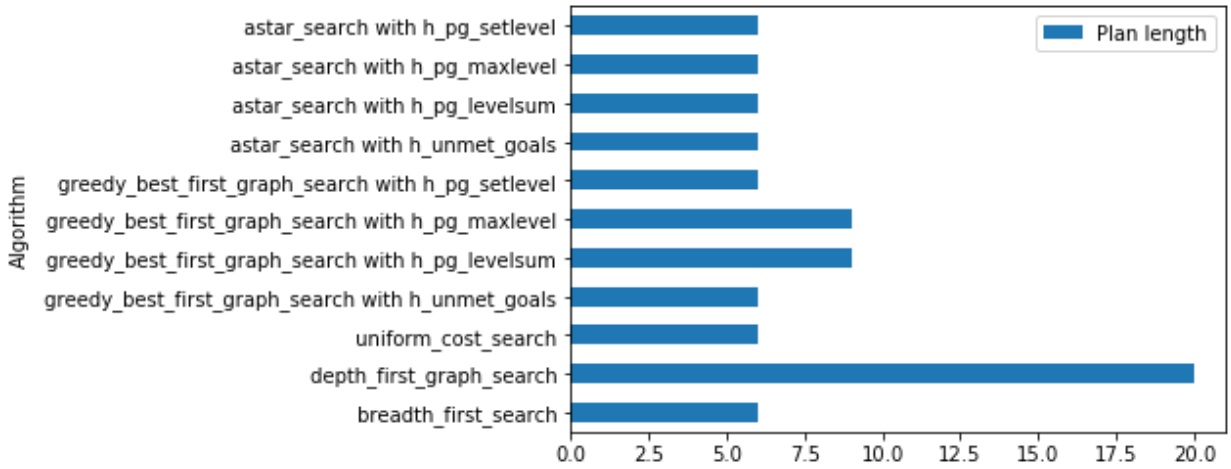
The table below shows the results of these search algorithms on Problems 1 and 2.

	Problem	Algorithm	Plan length	Time elapsed in seconds	# Actions	Expansions	Goal Tests	New Nodes
0	Problem 1	breadth_first_search	6	0.00633056	20	43	56	178
1	Problem 1	depth_first_graph_search	20	0.00356914	20	21	22	84
2	Problem 1	uniform_cost_search	6	0.0100346	20	60	62	240
3	Problem 1	greedy_best_first_graph_search with h_unmet_goals	6	0.0016154	20	7	9	29
4	Problem 1	greedy_best_first_graph_search with h_pg_levelsum	9	0.0297073	20	56	58	224
5	Problem 1	greedy_best_first_graph_search with h_pg_maxlevel	9	0.0311287	20	60	62	240
6	Problem 1	greedy_best_first_graph_search with h_pg_setlevel	6	0.667264	20	6	8	28
7	Problem 1	astar_search with h_unmet_goals	6	0.00969755	20	50	52	206
8	Problem 1	astar_search with h_pg_levelsum	6	0.0312676	20	60	62	240
9	Problem 1	astar_search with h_pg_maxlevel	6	0.0298494	20	60	62	240
10	Problem 1	astar_search with h_pg_setlevel	6	1.55202	20	33	35	138
11	Problem 2	breadth_first_search	9	2.06937	72	3343	4609	30503
12	Problem 2	depth_first_graph_search	619	3.1761	72	624	625	5602
13	Problem 2	uniform_cost_search	9	3.46456	72	5154	5156	46618
14	Problem 2	greedy_best_first_graph_search with h_unmet_goals	9	0.0193614	72	17	19	170
15	Problem 2	greedy_best_first_graph_search with h_pg_levelsum	16	19.251	72	5130	5132	46170
16	Problem 2	greedy_best_first_graph_search with h_pg_maxlevel	16	20.053	72	5589	5591	50301
17	Problem 2	greedy_best_first_graph_search with h_pg_setlevel	9	17.9638	72	9	11	84
18	Problem 2	astar_search with h_unmet_goals	9	2.29906	72	2467	2469	22522
19	Problem 2	astar_search with h_pg_levelsum	9	24.8735	72	5275	5277	47638
20	Problem 2	astar_search with h_pg_maxlevel	9	25.1447	72	5453	5455	49194
21	Problem 2	astar_search with h_pg_setlevel	9	1620.57	72	1037	1039	9605

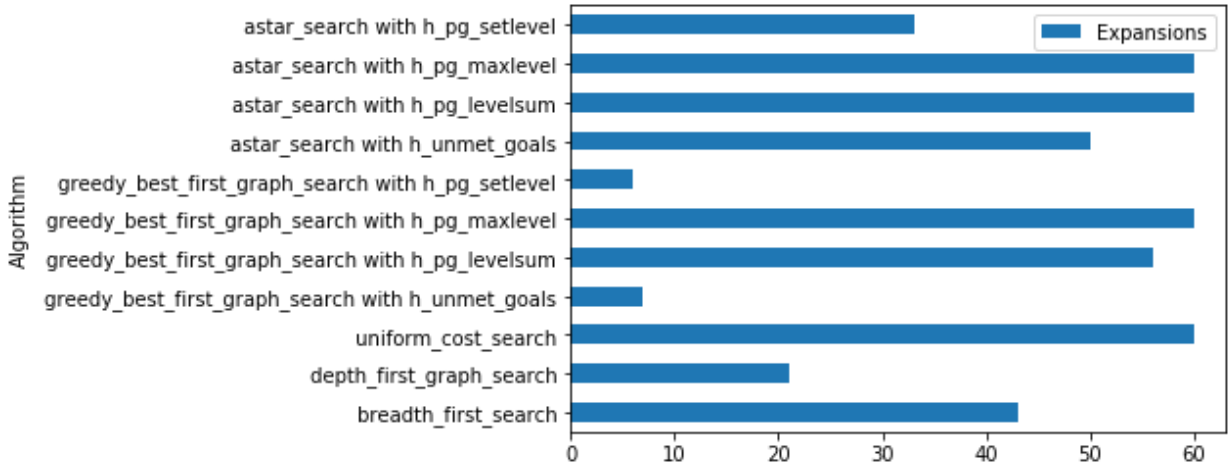
PROBLEM 1: TIME ELAPSED PER ALGORITHM



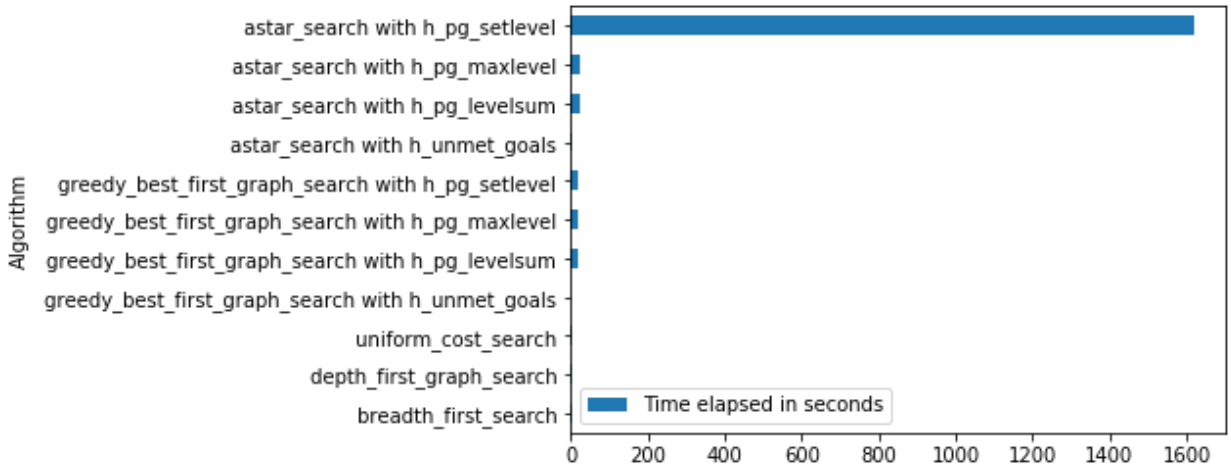
Problem 1: Plan Length



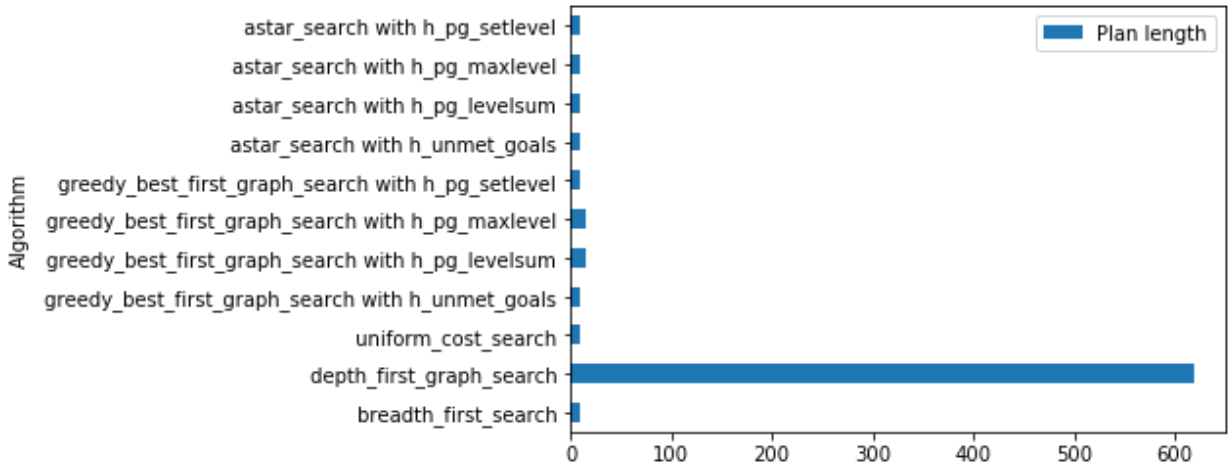
Problem1: Number of New Node Expansions



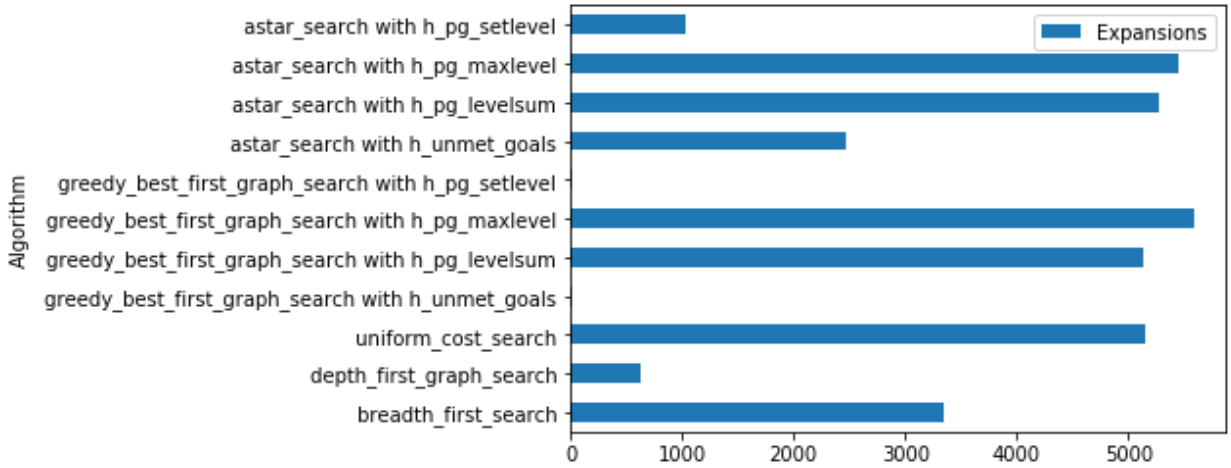
Problem 2: Time Elapsed per Algorithm



Problem 2: Plan Length



Problem 2: Number of New Node Expansions



Discussion

Based on the results from the table and charts above, we notice that the algorithm with the least amount of elapsed time is the Greedy Best First Graph Search with Unmet Goals Heuristic. This also seems to be the most efficient of the heuristic searches in general. Of the non-heuristic searches for Problem 1, Breadth First Search appears to be the most efficient since Plan Length is 6 with 43 expansions and time elapsed of 0.006s. For Problem 2 we also see the shortest Plan Length and elapsed time, but we notice that the Depth First Search performed poorly with a Plan Length of 619.

RESULTS OF PROBLEM 3 & 4

The following problems and search algorithms were run for the second run.

PROBLEMS:

1. Air Cargo Problem 3
2. Air Cargo Problem 4

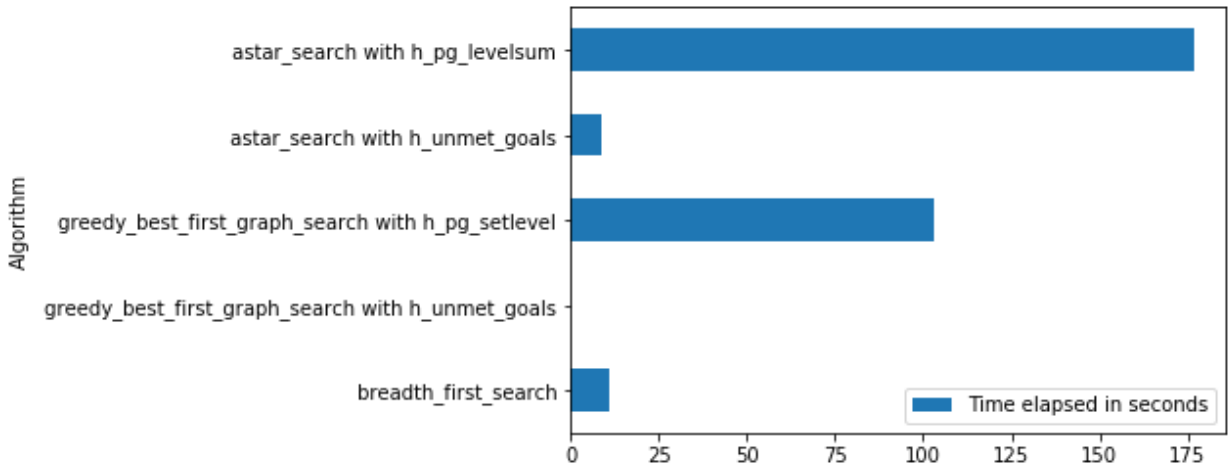
SEARCH ALGORITHMS:

1. breadth_first_search
2. greedy_best_first_graph_search h_unmet_goals
3. greedy_best_first_graph_search h_pg_setlevel
4. astar_search h_unmet_goals
5. astar_search h_pg_levelsum

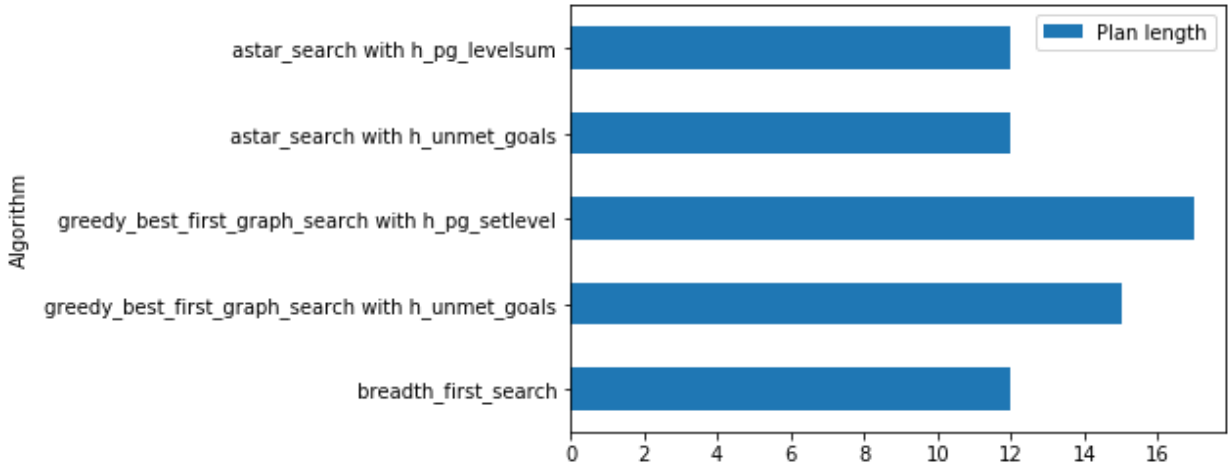
The table below shows the results of these search algorithms on Problems 3 and 4.

	Problem	Algorithm	Plan length	Time elapsed in seconds	# Actions	Expansions	Goal Tests	New Nodes
0	Problem 3	breadth_first_search	12	10.8742	88	14663	18098	129625
1	Problem 4	breadth_first_search	14	96.3507	104	99736	114953	944130
2	Problem 3	greedy_best_first_graph_search with h_unmet_goals	15	0.0375156	88	25	27	230
3	Problem 4	greedy_best_first_graph_search with h_unmet_goals	18	0.0619304	104	29	31	280
4	Problem 3	greedy_best_first_graph_search with h_pg_setlevel	17	102.824	88	35	37	345
5	Problem 4	greedy_best_first_graph_search with h_pg_setlevel	23	472.498	104	107	109	1164
6	Problem 3	astar_search with h_unmet_goals	12	8.69588	88	7388	7390	65711
7	Problem 4	astar_search with h_unmet_goals	14	57.2954	104	34330	34332	328509
8	Problem 3	astar_search with h_pg_levelsum	12	176.849	88	18848	18850	164668
9	Problem 4	astar_search with h_pg_levelsum	14	1581.6	104	114592	114594	1077277

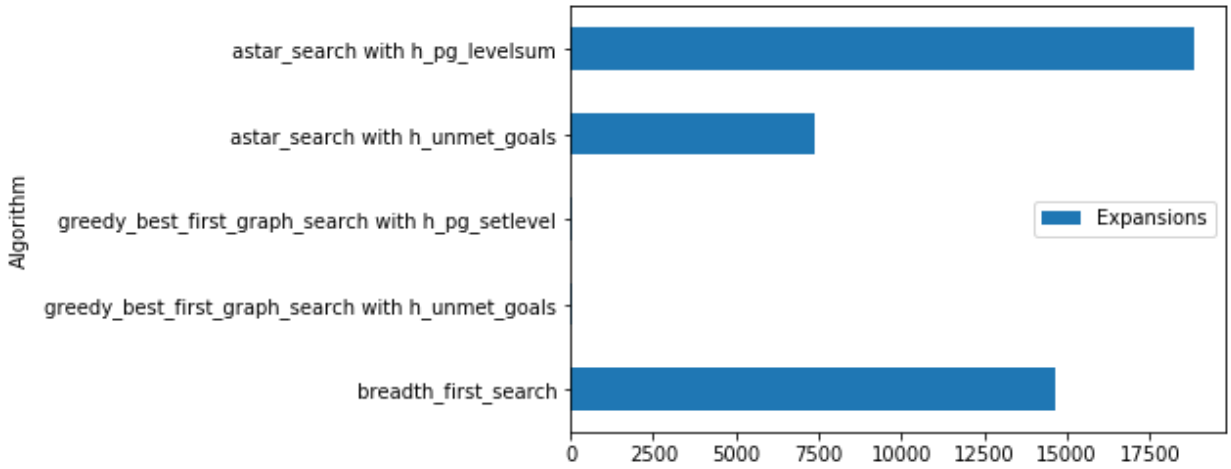
Problem 3: Time Elapsed per Algorithm



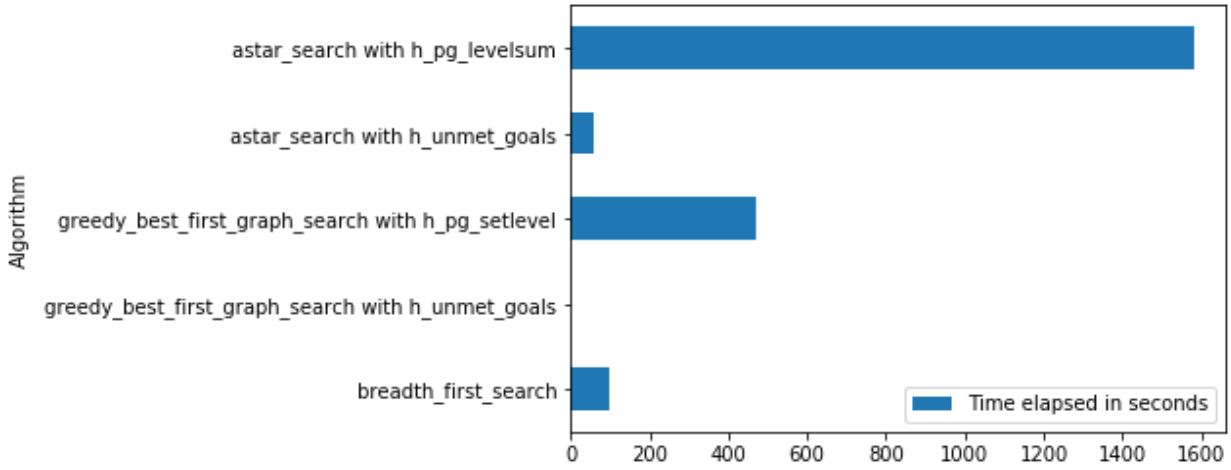
Problem 3: Plan Length



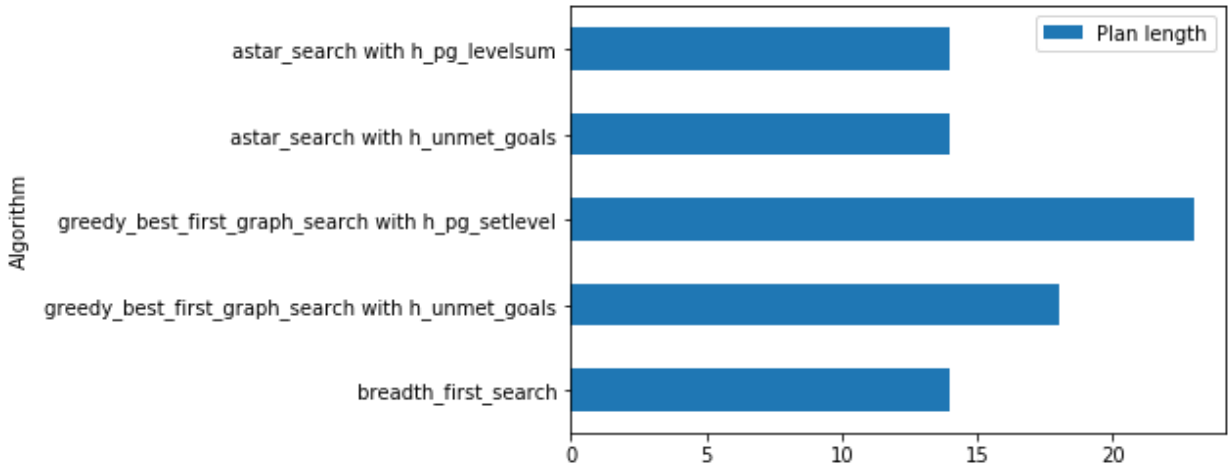
Problem 3: Number of New Node Expansions



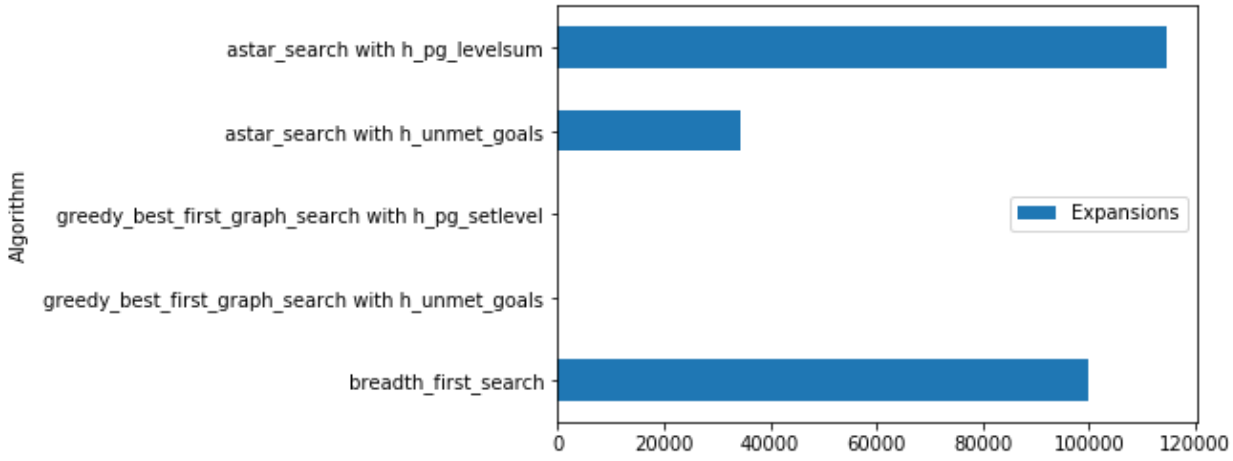
Problem 4: Time Elapsed per Algorithm



Problem 4: Plan Length



PROBLEM 4: NUMBER OF NEW NODE EXPANSIONS



DISCUSSION

Based on the table and charts above for Problems 3 and 4, we notice that the Greedy Best First Graph Search with Unmet Goals are the most time efficient, but not the most optimal in terms of Plan Length. The Breadth First Search and heuristic Astar Search with Unmet Goals produced the shortest Plan Length. It's also interesting to note that the Greedy Best First Graph Algorithms produced significantly lower New Nodes, Goal Tests and Expansions. The most optimal algorithm appears to be the Astar Search with Unmet Goals heuristic that has the lowest plan length, but longer elapsed time compared to the Greedy Best First Graph Algorithms. If time is a factor, the Greedy Best First Graph Algorithms would be a better choice of algorithm for these problems.

Algorithms that would be most appropriate for planning in a very restricted domain are the Greedy Best First Graph algorithms. This is due to the fact that they are fast and have fewer actions which are useful for real time applications. Algorithms more appropriate for planning very large domains would be the Breadth First and Astar search algorithms. We notice that the number of actions, expansions and new nodes increase with the complexity of the problem. Algorithms most suitable to find only optimal solutions are the search algorithms that are the most expensive in time elapsed, expansions and new nodes. The Astar search with level sum heuristic would likely be more suitable for searching for optimal solutions.