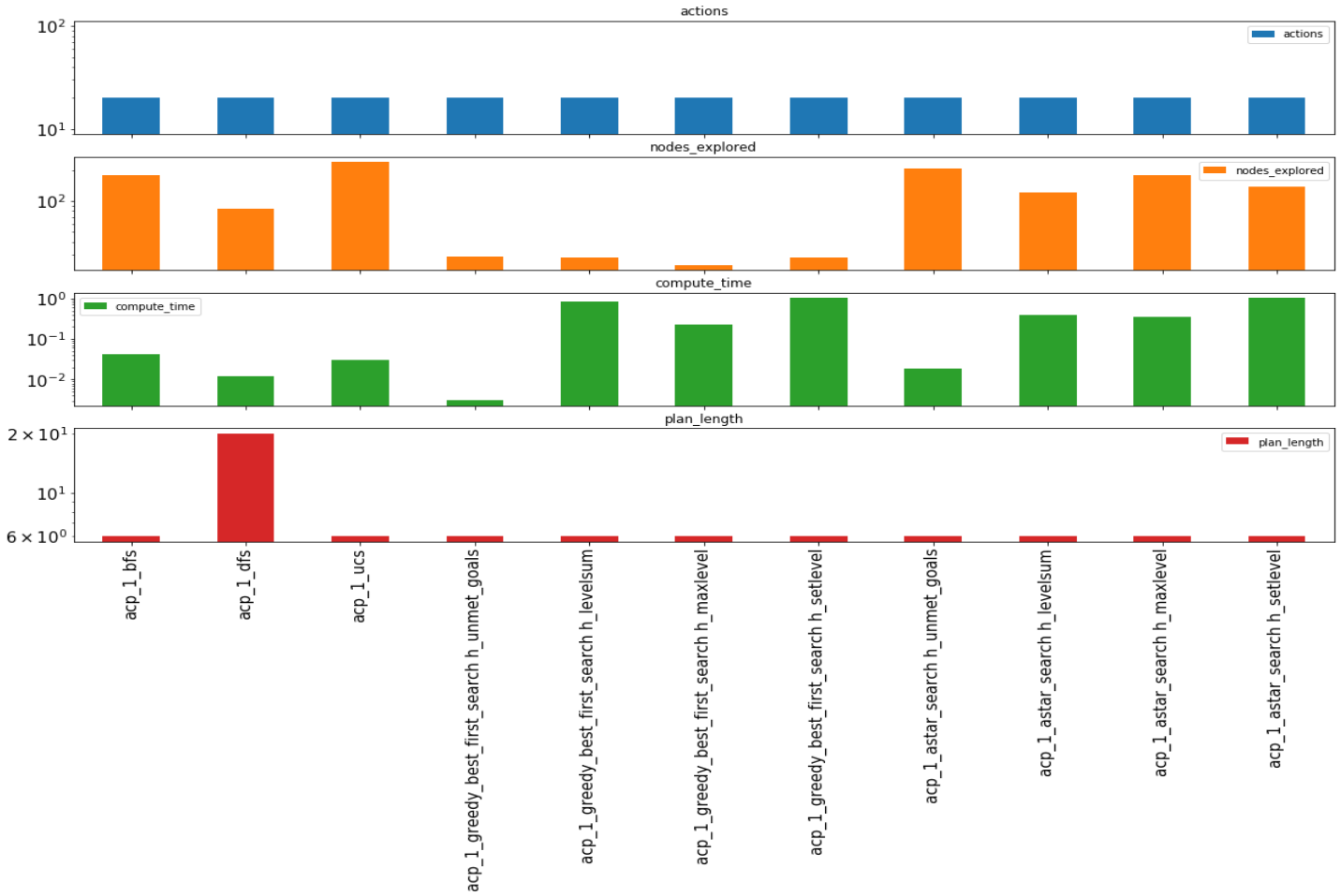
UDACITY PROJECT 2

Classical Planning

Kyle Topasna

Air Cargo Problem 1:

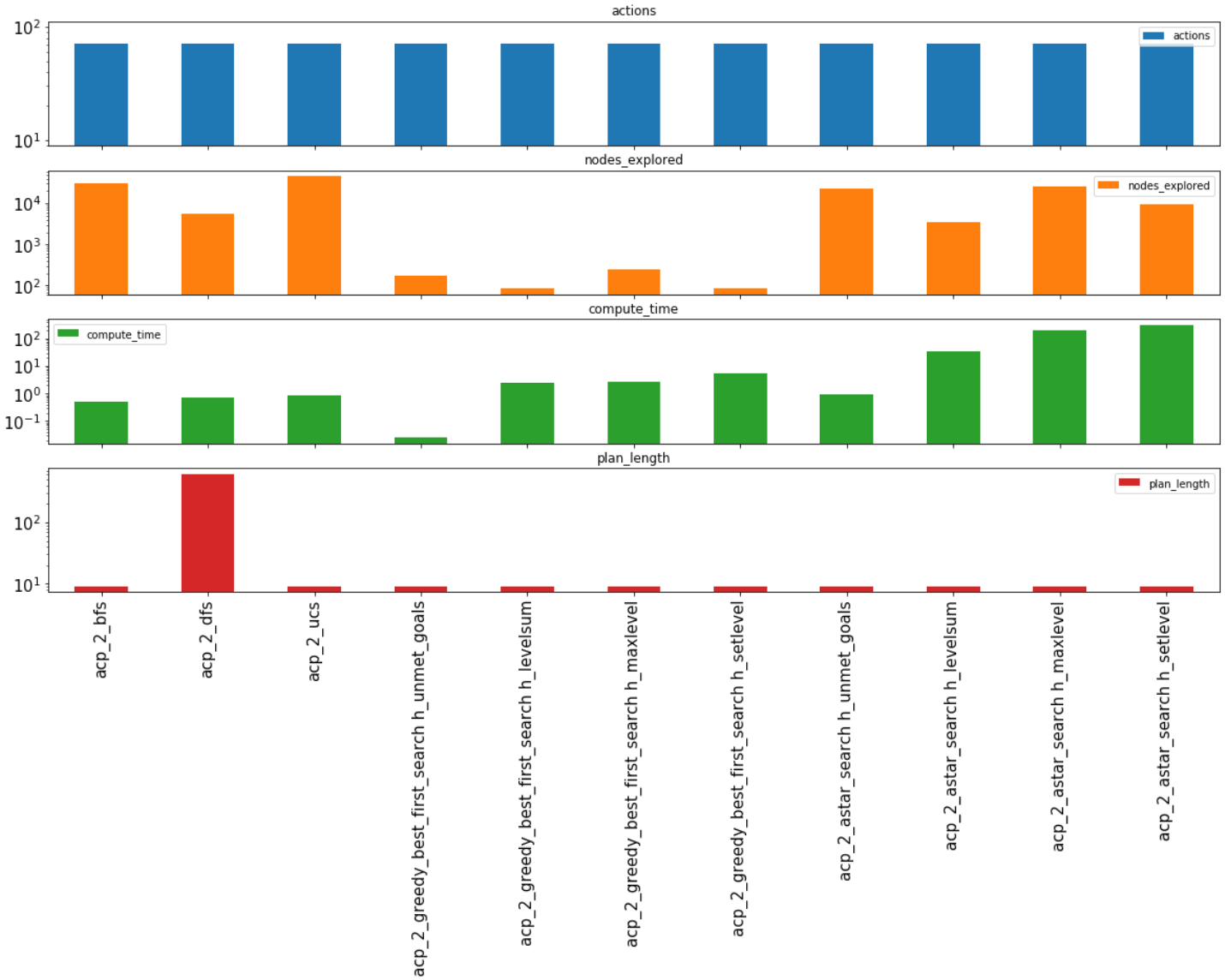
	actions	compute_time	graph_expansions	nodes_explored	plan_length
acp_1_bfs	20	0.042688	43	178	6
acp_1_dfs	20	0.012300	21	84	20
acp_1_ucs	20	0.030821	60	240	6
acp_1_greedy_best_first_search h_unmet_goals	20	0.003025	7	29	6
acp_1_greedy_best_first_search h_levelsum	20	0.826779	6	28	6
acp_1_greedy_best_first_search h_maxlevel	20	0.230910	6	24	6
acp_1_greedy_best_first_search h_setlevel	20	1.028256	6	28	6
acp_1_astar_search h_unmet_goals	20	0.019108	50	206	6
acp_1_astar_search h_levelsum	20	0.394338	28	122	6
acp_1_astar_search h_maxlevel	20	0.347104	43	180	6
acp_1_astar_search h_setlevel	20	1.048690	33	138	6



Air Cargo Problem 2:

	actions	compute_time	graph_expansions	nodes_explored	plan_length
acp_2_bfs	72	0.504165	3343	30503	9
acp_2_dfs	72	0.713412	624	5602	619
acp_2_ucs	72	0.893246	5154	46618	9
acp_2_greedy_best_first_search h_unmet_goals	72	0.024852	17	170	9
acp_2_greedy_best_first_search h_levelsum	72	2.571643	9	86	9
acp_2_greedy_best_first_search h_maxlevel	72	2.742849	27	249	9

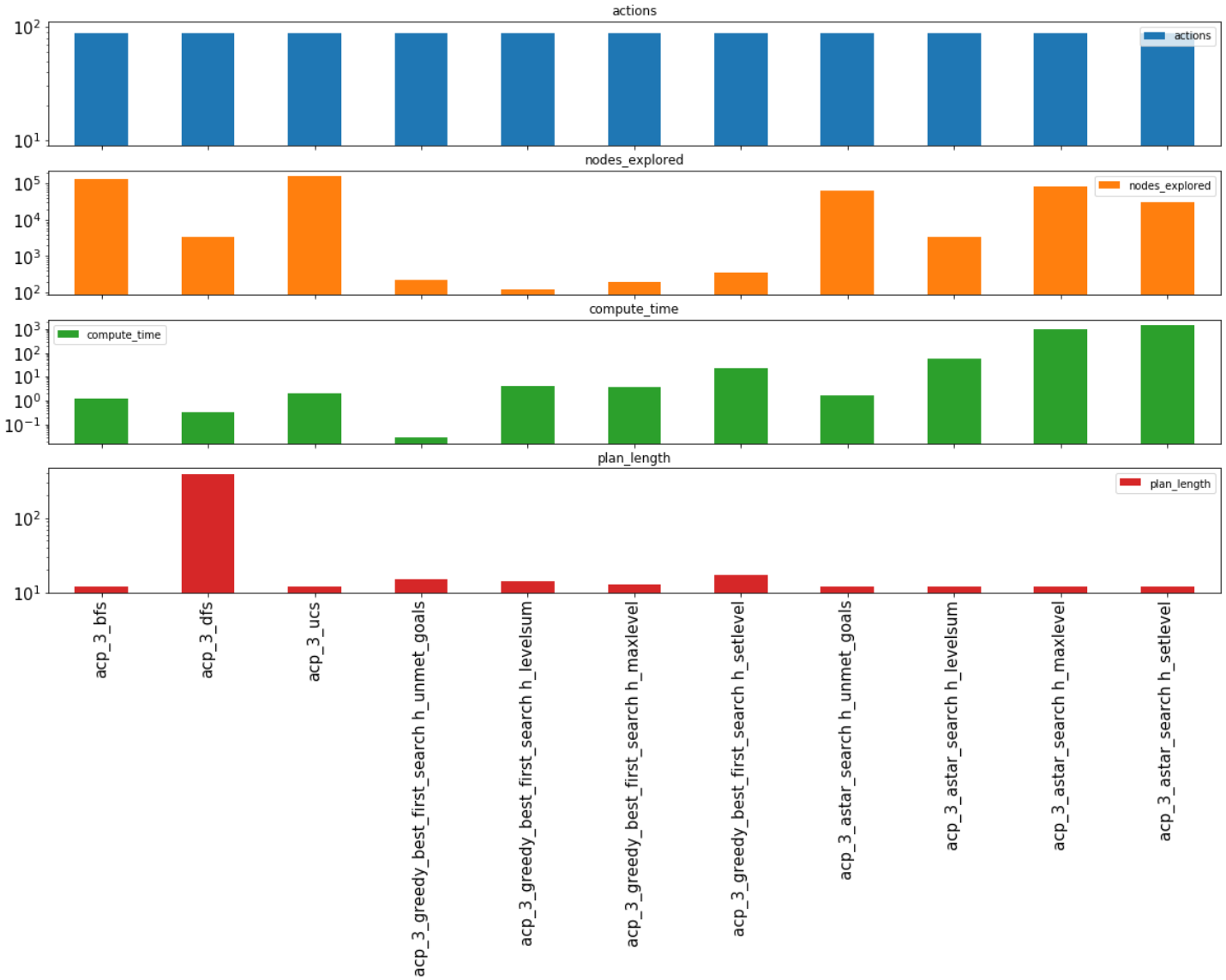
	actions	compute_time	graph_expansions	nodes_explored	plan_length
acp_2_greedy_best_first_search h_setlevel	72	5.399611	9	84	9
acp_2_astar_search h_unmet_goals	72	0.957883	2467	22522	9
acp_2_astar_search h_levelsum	72	35.475223	357	3426	9
acp_2_astar_search h_maxlevel	72	202.709443	2887	26594	9
acp_2_astar_search h_setlevel	72	322.180424	1037	9605	9



Air Cargo Problem 3:

	actions	compute_time	graph_expansions	nodes_explored	plan_length
acp_3_bfs	88	1.284745	14663	129625	12
acp_3_dfs	88	0.316636	408	3364	392
acp_3_ucs	88	1.957874	18510	161936	12
acp_3_greedy_best_first_search h_unmet_goals	88	0.027968	25	230	15
acp_3_greedy_best_first_search h_levelsum	88	4.171586	14	126	14
acp_3_greedy_best_first_search h_maxlevel	88	3.591809	21	195	13
acp_3_greedy_best_first_search h_setlevel	88	22.038052	35	345	17
acp_3_astar_search h_unmet_goals	88	1.705692	7388	65711	12
acp_3_astar_search h_levelsum	88	58.063234	369	3403	12
acp_3_astar_search h_maxlevel	88	937.574236	9580	86312	12
acp_3_astar_search h_setlevel	88	1470.351643	3423	31596	12

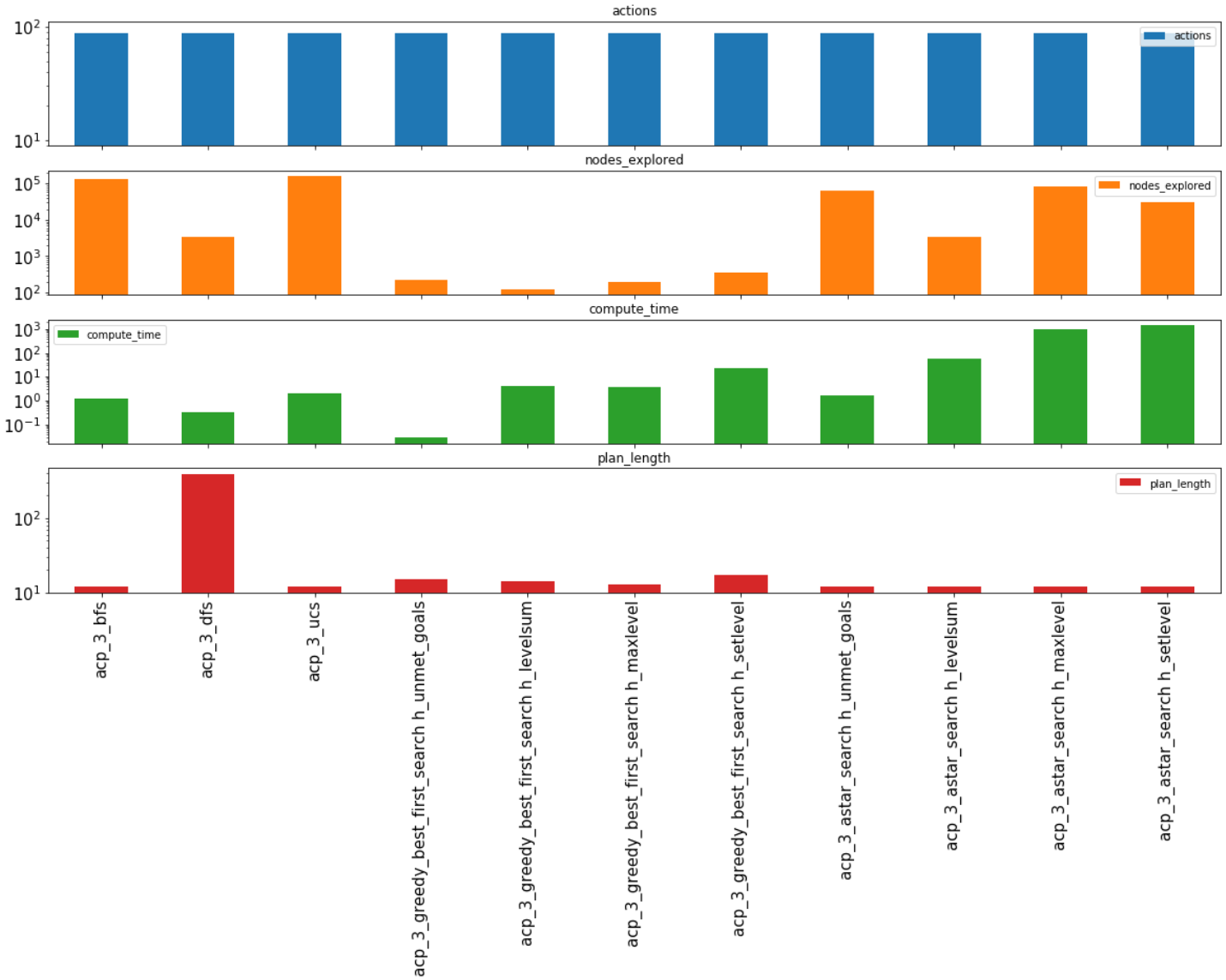
In [32]:



Air Cargo Problem 4:

	actions	compute_time	graph_expansions	nodes_explored	plan_length
acp_4_bfs	104	8.143862	99736	944130	14
acp_4_dfs	104	1730.883094	25174	228849	24132
acp_4_ucs	104	14.118177	113339	1066413	14
acp_4_greedy_best_first_search h_unmet_goals	104	0.067324	29	280	18

	actions	compute_time	graph_expansions	nodes_explored	plan_length
acp_4_greedy_best_first_search h_levelsum	104	7.887879	17	165	17
acp_4_greedy_best_first_search h_maxlevel	104	14.306673	56	580	17
acp_4_greedy_best_first_search h_setlevel	104	99.651816	107	1164	23
acp_4_astar_search h_unmet_goals	104	5.877387	34330	328509	14
acp_4_astar_search h_levelsum	104	345.657024	1208	12210	15
acp_4_astar_search h_maxlevel	104	9562.792889	62077	599376	14



Questions¶

Which algorithm or algorithms would be most appropriate for planning in a very restricted domain (i.e., one that has only a few actions) and needs to operate in real time?

Answer:

The greedy best first searches algorithms, regardless of heuristics, reliably performed the fast enough for real time operations when action states were kept low.

Which algorithm or algorithms would be most appropriate for planning in very large domains (e.g., planning delivery routes for all UPS drivers in the U.S. on a given day)

Answer:

Breadth First Search was the more optimal at exploring nodes and expanding the graph. A* searches worked well also at exploring large domains at the trade off for computational expense.

Which algorithm or algorithms would be most appropriate for planning problems where it is important to find only optimal plans?

Answer:

All A* Search variants and BFS reliably found the most optmial plan.