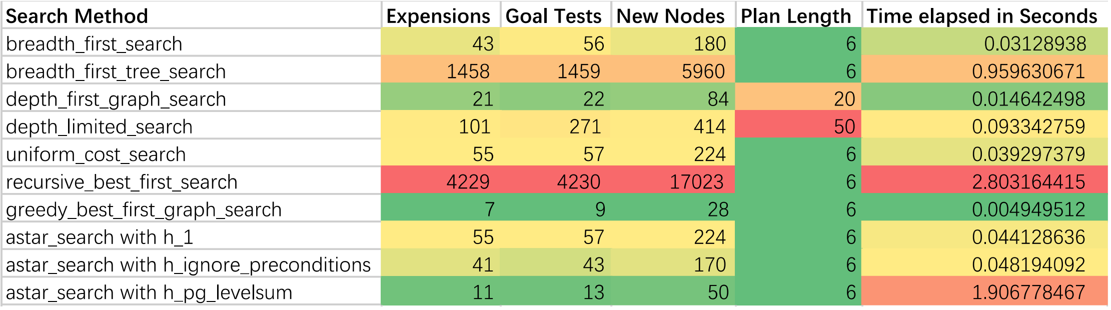
Heuristic Analysis

1. Result for each Problem

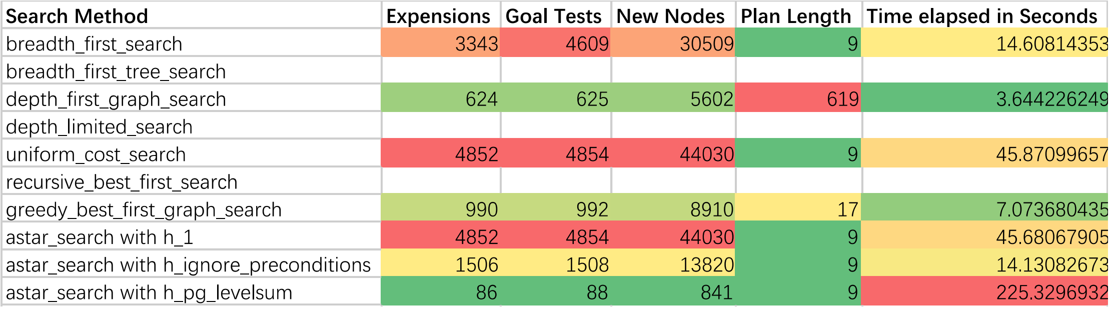
Problem 1



Search Method	Expensions	Goal Tests	New Nodes	Plan Length	Time elapsed in Seconds
breadth_first_search	43	56	180	6	0.03128938
breadth_first_tree_search	1458	1459	5960	6	0.959630671
depth_first_graph_search	21	22	84	20	0.014642498
depth_limited_search	101	271	414	50	0.093342759
uniform_cost_search	55	57	224	6	0.039297379
recursive_best_first_search	4229	4230	17023	6	2.803164415
greedy_best_first_graph_search	7	9	28	6	0.004949512
astar_search with h_1	55	57	224	6	0.044128636
astar_search with h_ignore_preconditions	41	43	170	6	0.048194092
astar_search with h_pg_levelsum	11	13	50	6	1.906778467

Optimal Plan:
Load(C1, P1, SFO)
Fly(P1, SFO, JFK)
Load(C2, P2, JFK)
Fly(P2, JFK, SFO)
Unload(C1, P1, JFK)
Unload(C2, P2, SFO)

Problem 2

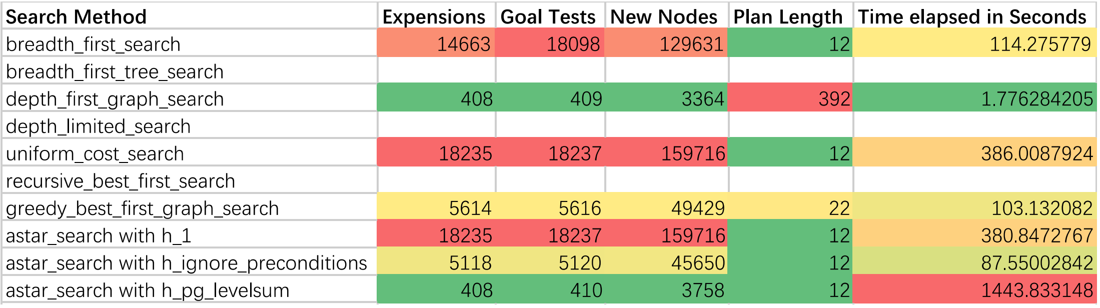


Search Method	Expensions	Goal Tests	New Nodes	Plan Length	Time elapsed in Seconds
breadth_first_search	3343	4609	30509	9	14.60814353
breadth_first_tree_search					
depth_first_graph_search	624	625	5602	619	3.644226249
depth_limited_search					
uniform_cost_search	4852	4854	44030	9	45.87099657
recursive_best_first_search					
greedy_best_first_graph_search	990	992	8910	17	7.073680435
astar_search with h_1	4852	4854	44030	9	45.68067905
astar_search with h_ignore_preconditions	1506	1508	13820	9	14.13082673
astar_search with h_pg_levelsum	86	88	841	9	225.3296932

Optimal Plan:
Load(C1, P1, SFO)
Fly(P1, SFO, JFK)
Load(C2, P2, JFK)
Fly(P2, JFK, SFO)
Load(C3, P3, ATL)
Fly(P3, ATL, SFO)

Unload(C3, P3, SFO)
Unload(C1, P1, JFK)
Unload(C2, P2, SFO)

Problem 3



Search Method	Expensions	Goal Tests	New Nodes	Plan Length	Time elapsed in Seconds
breadth_first_search	14663	18098	129631	12	114.275779
breadth_first_tree_search					
depth_first_graph_search	408	409	3364	392	1.776284205
depth_limited_search					
uniform_cost_search	18235	18237	159716	12	386.0087924
recursive_best_first_search					
greedy_best_first_graph_search	5614	5616	49429	22	103.132082
astar_search with h_1	18235	18237	159716	12	380.8472767
astar_search with h_ignore_preconditions	5118	5120	45650	12	87.55002842
astar_search with h_pg_levelsum	408	410	3758	12	1443.833148

Optimal Plan:
Load(C2, P2, JFK)
Fly(P2, JFK, ORD)
Load(C4, P2, ORD)
Fly(P2, ORD, SFO)
Load(C1, P1, SFO)
Fly(P1, SFO, ATL)
Load(C3, P1, ATL)
Fly(P1, ATL, JFK)
Unload(C4, P2, SFO)
Unload(C3, P1, JFK)
Unload(C1, P1, JFK)
Unload(C2, P2, SFO)

2. Analysis

Notice that in the above tables, I color coded the result so that green means good to make it more representative, and leave it blank if I cannot get a result in reasonable time. Overall, the search time increase rapidly as the problems getting more complex.

Among those non-heuristic search methods, depth first search does not guarantee the optimal result, which demonstrate the answer of the Quiz: Search Comparison in lesson 7. Most of time it will came up with a plan that the length is much longer than the optimal, it is unacceptable even though it can get a result fast. Uniform cost search and A-start search with h1 have the same number of expansions, goal tests and new nodes created, and almost same search time. Which is reasonable since they both based on best first graph search (according to AIMA code). It seems that breadth first search performs best among non-heuristic methods, it provides optimal plan in an acceptable time.

Both of our heuristic search method, A-star with h_ignore_preconditions and h_pg_sum produce optimal plan. Interesting fact is that with h_pg_sum, less node get created and expanded, shows that the heuristic guide the search better, but the overall runtime is longer. It is because the heuristic calculation is more time consuming.

Generally speaking, A-star search with h_ignore_preconditions is best for these problems. It searches less nodes than non-heuristic search and run faster, while still guarantee the optimal result.