Assignment

Run uninformed planning searches for air_cargo_p1, air_cargo_p2, and air_cargo_p3; provide metrics on number of node expansions required, number of goal tests, time elapsed, and optimality of solution for each search algorithm.

Run A planning searches using the heuristics you have implemented on air_cargo_p1, air_cargo_p2 and air_cargo_p3. Provide metrics on number of node expansions required, number of goal tests, time elapsed, and optimality of solution for each search algorithm and include the results in your report.*

Provide an optimal plan for Problems 1, 2, and 3.
Compare and contrast non-heuristic search result metrics (optimality, time elapsed, number of node expansions) for Problems 1,2, and 3. Include breadth-first, depth-first, and at least one other uninformed non-heuristic search in your comparison;

Compare and contrast heuristic search result metrics using A with the "ignore preconditions" and "level-sum" heuristics for Problems 1, 2, and 3.*
What was the best heuristic used in these problems? Was it better than non-heuristic search planning methods for all problems? Why or why not?

Result of the search result and metrics

(Problem1)

Search Method	#Nodes Expanded	# of goal tests	Time Elapsed (seconds)	Optimality
breadth_first_search	43	56	0.029	Yes
breadth_first_tree_search	1458	1459	0.601	Yes
depth_first_graph_search	12	13	0.009	No
depth_limited_search	101	271	0.065	No
uniform_cost_search	55	57	0.028	Yes
recursive_best_first_search h_1	4229	4230	1.623	Yes
greedy_best_first_graph_search h_1	7	9	0.004	Yes
astar_search h_1	55	57	0.027	Yes
astar_search h_ignore_preconditions	41	43	0.029	Yes
astar_search h_pg_levelsum	11	13	10.819	Yes

<non-heuristic search result of problem 1>
Breadth-first, bread-first-tree search, and uniform_cost_search could reach optimality. Depth-first-graph, and depth-limited search could not get optimality, even though their time elapsed was comparatively short.

<heuristic search result of problem 1>
Greedy-best-first performed best with minimum time elapsed, and nodes expanded. Other heuristics

required more time elapsed or nodes expanded, though all of them could reach optimality.
In terms of astar-group, ignore_precodition heuristic seemed best in total, because its time elapsed was shorter or almost same to others, and the number of node expanded was also reasonable. H-pg-levelsum heuristic got good performance for the number of node expanded, but it took very longer time than others.

<All search result of problem 1>
In terms of problem1, heuristic, greedy-best-first search performed the best, though this method could not get optimality in the problem2 & 3, as written as below. It seems that problem1 is simple enough so that greedy-best-first could work.

(Problem2)

Search Method	#Nodes Expanded	# of goal tests	Time Elapsed (seconds)	Optimality
breadth_first_search	3343	4609	7.092	Yes
breadth_first_tree_search	n/a	n/a	n/a	n/a
depth_first_graph_search	476	477	2.183	No
depth_limited_search	n/a	n/a	n/a	n/a
uniform_cost_search	4852	4854	7.945	Yes
recursive_best_first_search h_1	n/a	n/a	n/a	n/a
greedy_best_first_graph_search h_1	990	992	1.636	No
astar_search h_1	4852	4854	8.327	Yes
astar_search h_ignore_preconditions	1450	1452	3.174	Yes
astar_search h_pg_levelsum	n/a	n/a	n/a	n/a

<non-heuristic search result of problem 2>
Breadth-first, and uniform_cost_search could reach optimality. Breadth-first-tree, Depth-first-graph, and depth-limited search could not get optimality.

<heuristic search result of problem 2>
Neither greedy_best_first performed could reach optimality in this problem, nor h-pg-levelsum heuristic with astar. H-1 and h-ignore-preconditions got optimality, and the latter one achieved best result from every matrics.

<All search result of problem 2>
In terms of problem2, heuristic, astar with ignore-precondition heuristic performed the best. Breadth-first- search or astar, which is adding heuristic with breadth-first-search, only could work, and heuristic had some effect to optimize solving the problem.

(Problem3)

Search Method	#Nodes Expanded	# of goal tests	Time Elapsed (seconds)	Optimality
breadth_first_search	14663	18098	30.473	Yes
breadth_first_tree_search	n/a	n/a	n/a	n/a
depth_first_graph_search	1511	1512	11.302	No
depth_limited_search	n/a	n/a	n/a	n/a
uniform_cost_search	18223	18225	36.811	Yes
recursive_best_first_search h_1	n/a	n/a	n/a	n/a
greedy_best_first_graph_search h_1	5580	5582	11.500	No
astar_search h_1	18223	18225	36.394	Yes
astar_search h_ignore_preconditions	5040	5042	13.140	Yes
astar_search h_pg_levelsum	n/a	n/a	n/a	n/a

<non-heuristic search result of problem 3>
Breadth-first, and uniform_cost_search could reach optimality. Breadth-first-tree, Depth-first-graph, and depth-limited search could not get optimality.

<heuristic search result of problem 3>
Neither greedy_best_first performed could reach optimality in this problem, nor h-pg-levelsum heuristic with astar. H-1 and h-ignore-preconditions got optimality, and the latter one achieved best result from every matrics.

<All search result of problem 3>
In terms of problem3, the result was almost same as the result of problem2, intrinsically. Planning Graphs works well for solving the problem widely, and heuristic seemed necessary to make it more effective.

(End)