Written Analysis

1) Optimal plan for Problem 1, 2 and 3

a. Problem #1

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

b. Problem #2

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

c. Problem #3

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

2) Compare and contrast non-heuristic search result metrics

a. Problem #1

	Plan Length	Expansions	Goal Test	New Nodes	Elapsed Time
breadth_first_search	6	43	56	180	0.0266
depth_first_graph_search	12	12	13	48	0.0068
uniform_cost_search	6	55	57	224	0.0324

b. Problem #2

	Plan Length	Expansions	Goal Test	New Nodes	Elapsed Time
breadth_first_search	9	3343	4609	30509	11.9371
depth_first_graph_search	1444	1669	1670	14863	11.4688
uniform_cost_search	9	4853	4855	44041	10.4947

c. Problem #3

	Plan Length	Expansions	Goal Test	New Nodes	Elapsed Time
breadth_first_search	12	14663	18098	129631	96.1615
depth_first_graph_search	571	592	593	4927	2.6609
uniform_cost_search	12	18164	18166	159147	48.0912

Comment:

i. Optimality

Both breadth first and uniform cost search come up with a very good optimality with small plan length, while the depth first search shows a very huge plan length. Just like in the video lesson mentions, since unlimited depth search keeps searching one branch until it runs out of actions or until it reaches all goal, so it makes sense to see it searches very deep in one branch and comes up with a very long non-optimal plan. At the meantime breath first and uniform search show a similar optimality.

ii. Time elapsed

In term of time elapsed, depth first seems much faster overall. But since depth first searches the deepest node first, sometime it gets lucky and reach the goals very early, and sometimes does not. The relative elapsed time of depth first search varies. At the meantime, when comparing breath first search and uniform cost search, I notice breath first is faster for smaller problem (compare to uniform), while uniform search become relatively mush faster when the problem getting bigger.

iii. Number of Nodes

Overall, uniform cost search search highest number of nodes, while breath first search seconds it, and depth first search searches much less number of nodes.

3) Compare and contrast heuristic search result metrics

a. Problem #1

		Plan Length	Expansions	Goal Test	New Nodes	Elapsed Time
A*	h_ignore_preconditions	6	55	57	224	0.0347
A*	h_pg_levelsum	6	19	21	86	0.8132

b. Problem #2

		Plan Length	Expansions	Goal Test	New Nodes	Elapsed Time
A*	h_ignore_preconditions	9	4853	4855	44041	10.8681
A*	h_pg_levelsum	9	1123	1125	10337	315.0008

c. Problem #3

		Plan Length	Expansions	Goal Test	New Nodes	Elapsed Time
A*	h_ignore_preconditions	12	18164	18166	159147	50.522
A*	h_pg_levelsum	13	4100	4102	35993	1512.1437

Comment:

i. Optimality

In term of optimality, ignore conditions and level sum are tied. However, ignore precondition heuristic may be slightly more optimal than level sum since it comes up with a plan length of 12, which is slightly better than the 13 of level sum. But the difference in this test is so small, so this might just be an coincident.

ii. Time elapsed

In term of elapsed time, the ignore preconditions is significantly faster.

iii. Number of Nodes

While the ignore preconditions heuristic is faster, it also visits more nodes than level sum. This could be result of the higher complexity for computing level sum than ignore condition heuristic.

4) Best heuristic used in these problems

I. Overall for this particular set of problems, the ignore preconditions heuristic is the best heuristic. It almost comes with the best elapsed time (especially when the problem is more complicated), and the most optimal plan. Also it visits the most nodes so that it may be guaranteed to find the best plan.

II. It is better than non-heuristic search planning method except for the very simple problem 1. In a simple problem like problem 1, its elapsed time is slightly higher than the breath first search.