1. Optimal plans

Problem 1

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

Problem 2

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

Problem 3

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

2. Search results

Three uninformed search methods (breadth-first search, depth-first graph search and uniform cost search) have been implemented. As for the informed search methods, A* search with h1, A* search with ignore preconditions and A*search with level sum have been performed. The table below summarizes results obtained from various non-heuristic and heuristic search planning methods:

Problem	Search	Heuristic	Expansions	Goal tests	Path length	Time Elapsed (seconds)	Optimal (Y/N)
1	Breadth-first	-	43	56	6	0.055	Yes
	Depth-first graph	-	21	22	20	0.046	No
	Uniform-cost	-	55	57	6	0.062	Yes
	A*	h1	55	57	6	0.065	Yes
	A*	Ignore preconditions	41	43	6	0.046	Yes
	A*	Level-sum	11	13	6	4.880	Yes
2	Breadth-first	-	3343	4609	9	19.954	Yes
	Depth-first graph	-	624	625	619	4.994	No
	Uniform-cost	-	4852	4854	9	70.325	Yes
	A*	h1	4852	4854	9	70.178	Yes
	A*	Ignore preconditions	1506	1508	9	19.424	Yes
	A*	Level-sum	-	-	-	-	-
3	Breadth-first	-	14663	18098	12	155.670	Yes
	Depth-first graph	-	408	409	392	2.626	No
	Uniform-cost	-	-	-	-	-	-
	A*	h1	-	-	-	-	-
	A*	Ignore preconditions	5118	5120	12	137.867	Yes
	A*	Level-sum	-	-	-	-	-

- *Note that the following results have been omitted as their execution times exceeded 10 minutes:*
  Problem 2: A with level-sum heuristic,*
  Problem 3: uniform-cost search, A with h1 heuristic and A* with level-sum heuristic.*
- *It should be clarified that all of heuristic search methods yielded optimal path lengths, even though data for some of them are not shown due to the aforementioned long runtime issue.*

3. Analysis

Comparison: non-heuristic search result metrics

The optimal path is the one with the shortest path length – 6 for problem 1, 9 for problem 2 and 12 for problem 3. Therefore, according to the results, both breadth-first search and uniform-cost search are optimal for all three problems. It should be noted that uniform-cost search took longer than 10 minutes to execute in problem 3. This was probably due to the fact that the algorithm had to explore options in every direction, which added to the computation time. In terms of speed, depth-first graph search outperformed the other two methods and expanded far fewer nodes (used less memory), but failed to give an optimal action plan in any of the problems. The failure to was probably because depth-first search is neither a complete or optimal search strategy (section 3.5, Artificial Intelligence: A Modern Approach). If finding the right path quickly is more important than finding the shortest path

to the goal then depth-first graph search is recommended. This is especially true for more complicated problems like problem 3, where depth-first graph search was 50 times and 250 times faster than breadth-first search and uniform-cost search, respectively. Otherwise, breadth-first search would be the best choice due to its ability to find optimal paths and its higher speed compared to uniform-cost search.

Comparison: automatic heuristics

All of the tested heuristic strategies returned optimal action plans for all problems. However, A* with level sum exceeded 10 minutes of execution time in both problems 2 and 3. Similarly, A* with h1 heuristic failed to generate an action plan in under the time limit. The long time required by A* search with level-sum heuristic was likely due to its reliance on GraphPlan, which adds to the time to compute the heuristic. In contrast, A* search with ignore preconditions heuristic consistently generated optimal search plans in all problems. Nevertheless, the level-sum heuristic required considerably fewer node expansions (in problem 1) than the other two heuristic methods, suggesting that it used less memory to execute.

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?

The best heuristic used in these problems was A* search with ignore preconditions heuristic. This is because only heuristic yielded an optimal action plan in under 10 minutes in all three problems. The ignore preconditions heuristic was better than all non-heuristic planning methods in all problems because it outperformed breadth-first search in speed and depth-first graph search in optimality. Furthermore, the ignore preconditions heuristic used less memory than breadth-first search in all problems. Therefore, the ignore preconditions heuristic has the appropriate balance of speed and optimality that makes it better than non-heuristic methods and other tested heuristics.