Problem1	breadth_first	depth_first	uniform_cost	h_ignore_pre	astar_search
Time(s)	0.0323	0.0086	0.0410	0.0447	3.2470
Plan length	6	12	6	6	6
Expansion	Expansions 43 Goal Tests 56 New Nodes 180	Expansions 2 Goal Tests 13 New Nodes 48	Expansions 55 Goal Tests 57 New Nodes 224	Expansions 41 Goal Tests 43 New Nodes 170	Expansions 41 Goal Tests 43 New Nodes 170
Plan	Load(C2, P2, JFK) Load(C1, P1, SFO) Fly(P2, JFK, SFO) Unload(C2, P2, SFO) Fly(P1, SFO, JFK) Unload(C1, P1, JFK)	Fly(P1, SFO, JFK) Fly(P2, JFK, SFO) Load(C1, P2, SFO) Fly(P2, SFO, JFK) Fly(P1, JFK, SFO) Unload(C1, P2, JFK) Fly(P2, JFK, SFO) Fly(P1, SFO, JFK) Load(C2, P1, JFK) Fly(P2, SFO, JFK) Fly(P1, JFK, SFO) Unload(C2, P1, SFO)	Load(C1, P1, SFO) Load(C2, P2, JFK) Fly(P1, SFO, JFK) Fly(P2, JFK, SFO) Unload(C1, P1, JFK) Unload(C2, P2, SFO)	Load(C1, P1, SFO) Load(C2, P2, JFK) Fly(P1, SFO, JFK) Unload(C1, P1, JFK) Fly(P2, JFK, SFO) Unload(C2, P2, SFO)	Load(C1, P1, SFO) Fly(P1, SFO, JFK) Unload(C1, P1, JFK) Load(C2, P2, JFK) Fly(P2, JFK, SFO) Unload(C2, P2, SFO)
Optimal	Yes	No	Yes	Yes	Yes

Problem2	breadth_first	depth_first	uniform_cost	h_ignore_pre	astar_search
Time(s)	11.99	12.4	35.93	11.2	1476.5
Plan length	9	1444	9	9	9
Expansions	Expansions 3343 Goal Tests 4609 New Nodes 30509	Expansions 1669 Goal Tests 1670 New Nodes 14863	Expansions 4852 Goal Tests 4854 New Nodes 44030	Expansions 1506 Goal Tests 1508 New Nodes 13820	Expansions 1245 Goal Tests 1247 New Nodes 11307
Plans	Load(C2, P2, JFK) Load(C1, P1, SFO) 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)	Too long to put here…	Load(C1, P1, SFO) Load(C2, P2, JFK) Load(C3, P3, ATL) Fly(P1, SFO, JFK) Fly(P2, JFK, SFO) Fly(P3, ATL, SFO) Unload(C3, P3, SFO) Unload(C1, P1, JFK) Unload(C2, P2, SFO)	Load(C3, P3, ATL) Fly(P3, ATL, SFO) Unload(C3, P3, SFO) Load(C1, P1, SFO) Fly(P1, SFO, JFK) Unload(C1, P1, JFK) Load(C2, P2, JFK) Fly(P2, JFK, SFO) Unload(C2, P2, SFO)	Load(C3, P3, ATL) Fly(P3, ATL, SFO) Unload(C3, P3, SFO) Load(C1, P1, SFO) Fly(P1, SFO, JFK) Unload(C1, P1, JFK) Load(C2, P2, JFK) Fly(P2, JFK, SFO) Unload(C2, P2, SFO)
Optimal	Yes	No	Yes	Yes	Yes

Problem3	breadth_first	depth_first	uniform_cost	h_ignore_pre	astar_search
Time(s)	87.11	2.538	297.5	68.359	5105
Plan length	12	571	12	12	12
Expansions	Expansions 663 Goal Tests 18098 New Nodes 129631	Expansions 592 Goal Tests 593 New Nodes 4927	Expansions 18235 Goal Tests 18237 New Nodes 159716	Expansions 5118 Goal Tests 5120 New Nodes 45650	Expansions 2934 Goal Tests 2936 New Nodes 26122
Plans	Load(C2, P2, JFK) Load(C1, P1, SFO) Fly(P2, JFK, ORD) Load(C4, P2, ORD) Fly(P1, SFO, ATL) Load(C3, P1, ATL) Fly(P1, ATL, JFK) Unload(C1, P1, JFK) Unload(C3, P1, JFK) Fly(P2, ORD, SFO) Unload(C2, P2, SFO) Unload(C4, P2, SFO)	Too long to put here	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(P2, ORD, SFO) Fly(P1, ATL, JFK) Unload(C4, P2, SFO) Unload(C3, P1, JFK) Unload(C1, P1, JFK) Unload(C2, P2, SFO)	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(C1, P1, JFK) Unload(C2, P2, SFO)	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(P2, ORD, SFO) Unload(C4, P2, SFO) Fly(P1, ATL, JFK) Unload(C3, P1, JFK) Unload(C1, P1, JFK) Unload(C2, P2, SFO)
Optimal	Yes	No	Yes	Yes	Yes

All problems	breadth_first	depth_first	uniform_cost	h_ignore_pre	astar_search
Total time:	99.4	14.9	333.4	79.54	6584

Performance summary:

In my report I compared breath first search, depth first graph search, uniform cost research, and a start search using two different heuristics, ignore preconditions and level sum heuristic. Overall all algorithms except depth first search gave me the optimal plan with the shortest length. As explained in the videos, the reason why the depth first search fails to find the optimal path is that it always explores the deepest branches and stops as soon as it finds a goal. In this case we have essentially an unlimited tree because we can always keep doing a different actions, so depth first search can keep on going forever. I still think that this algorithm can be useful in a case where we want to know whether there is a path to the goal.
In terms of the time performance of optimal algorithms a star search with ignore preconditions was the best with the shortest total time across all three problems. I think this is expected because a*star is like breath first search with the addition of a heuristic that allows us to guess the total cost of a path. So instead of expanding from the initial state in circles we expand towards to the goal according to the heuristic.
In terms of expanding the least number of nodes a star search with level sum heuristic was the best among the optimal algorithms. What this tell us is that level sum heuristic is more accurate than ignore preconditions heuristic and better in guiding us toward the goal. However because calculating level sum heuristic was very slow, we ended up making a huge sacrifice in total execution time. I think this algorithm would still be preferred in situations where it takes more time to expand a node than to calculate the heuristic, like in real life when expanding a node corresponds to an important decision like making a career change.
Overall, 4 of the 5 algorithm gave the optimal plan, a star search with ignore preconditions had the fastest time and a star with level sum had the last number of nodes expanded.