# Optimal Plan

**Problem 1:**

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:**

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(C2, P2, SFO)

Unload(C1, P1, JFK)

**Problem 3:**

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(C2, P2, SFO)

Unload(C1, P1, JFK)

# Search Comparison

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Optimality(Length) | | | Time elapsed | | | Node expansions | | |
| **Problem 1** | **Problem 2** | **Problem 3** | **Problem 1** | **Problem 2** | **Problem 3** | **Problem 1** | **Problem 2** | **Problem 3** |
| Breadth first | 6 | 9 | 12 | 0.02 | 11.55 | 92.70 | 43 | 3343 | 14663 |
| Depth first | 12 | 575 | 596 | 0.007 | 3.00 | 3.00 | 12 | 582 | 627 |
| Uniform cost | 6 | 9 | 12 | 0.03 | 10.12 | 39.40 | 55 | 4853 | 18223 |
| A\* ignore precond | 6 | 9 | 12 | 0.03 | 3.31 | 12.54 | 41 | 1450 | 5040 |
| A\* levelsum | 6 | 9 | 12 | 0.67 | 52.92 | 272.16 | 11 | 86 | 325 |

Breadth First Search, Depth First Search, A\* Search with ignore preconditions heuristic and A\* Search with level sum heuristic all return with optimal plan in these three cases. **Depth First Search doesn’t come up with an optimal search because it didn’t put the length of actions as a priority. It follows one path until reaches the goal. Not like Breadth First Search, where steps were automatically considered into the algorithm, Depth First Search doesn’t consider if the goal comes earlier.**

A\* Search with ignore preconditions heuristic gives the optimal plan. It takes less time than the other search algorithms that come up with optimal plans. And it expands less nodes that the non-heuristic search algorithms. It keeps a balance between time spent and node expanded. **The reason why ignore preconditions heuristic takes less time is that it drops all preconditions. That makes every action applicable in every state and in one step, which almost implies the number of steps required to solve the relaxed problem is the number of unsatisfied goals.**

A\* Search with level sum heuristic ends up the longest time among these four algorithms. **Because it needs to spend time creating the planning graph every time a node is expanded.** **Level sum heuristic could be admissible, but works well in practice for problems that are largely decomposable.** But it explores the least number of nodes during search. This heuristic traded time for optimality and times of exploring nodes.

In my opinion, ignore preconditions heuristic is the best heuristic among the heuristics used in this project. It spends reasonable time and explores reasonable number of nodes. And it takes less memory space than level sum heuristic since level sum heuristic need to draw a planning graph each time a node is expanded. For a more complicated problem, level sum heuristic might win over ignore preconditions heuristic because it explores less nodes.

*Artificial Intelligence: A Modern Approach by Norvig and Russell*