# Heuristic Analysis

## Result for each Problem

### Problem 1



Optimal Plan:

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



Optimal Plan:

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(C1, P1, JFK)

Unload(C2, P2, SFO)

### Problem 3



Optimal Plan:

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(C1, P1, JFK)

Unload(C2, P2, SFO)

## Analysis

Notice that in the above tables, I color coded the result so that green means good to make it more representative, and leave it blank if I cannot get a result in reasonable time. Overall, the search time increase rapidly as the problems getting more complex.

Among those non-heuristic search methods, depth first search does not guarantee the optimal result, which demonstrate the answer of the Quiz: Search Comparison in lesson 7. Most of time it will came up with a plan that the length is much longer than the optimal, it is unacceptable even though it can get a result fast. Uniform cost search and A-start search with h1 have the same number of expansions, goal tests and new nodes created, and almost same search time. Which is reasonable since they both based on best first graph search (according to AIMA code). It seems that breadth first search performs best among non-heuristic methods, it provides optimal plan in an acceptable time.

Both of our heuristic search method, A-star with h\_ignore\_preconditions and h\_pg\_sum produce optimal plan. Interesting fact is that with h\_pg\_sum, less node get created and expanded, shows that the heuristic guide the search better, but the overall runtime is longer. It is because the heuristic calculation is more time consuming.

Generally speaking, A-star search with h\_ignore\_preconditions is best for these problems. It searches less nodes than non-heuristic search and run faster, while still guarantee the optimal result.