Heuristic Review

1. Provide an optimal plan for Problems 1, 2, and 3.

For Problem 1, an optimal plan would be:
Load(C1, P1, SFO)
Load(C2, P2, JFK)
Fly(P1, SFO, JFK)
Fly(P2, JFK, SFO)
Unload(C1, P1, JFK)
Unload(C2, P2, SFO)

For Problem 2, an optimal plan would be:
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)

For Problem 3, an optimal plan would be:
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(C2, P2, SFO)
Unload(C1, P1, JFK)

2. Compare and contrast non-heuristic search result metrics (optimality, time elapsed, number of node expansions) for Problems 1,2, and 3. Include breadth-first, depth-first, and at least one other uninformed non-heuristic search in your comparison.

	S1(BFS) Exp/ GoalTest/ NewNode – Time elapsed	S3(DFS) Exp/ GoalTest/ NewNode – Time elapsed	S5(Uniform) Exp/ GoalTest/ NewNode – Time elapsed
P1	43/56/180 – 0.062s	12/13/48 – 0.0092s	55/57/224 – 0.062s
P2	3343/4609/30509 – 25.67s	582/583/5211 – 43.04s	4853/4855/44041 – 17.74s
P3	14663/18098/129631 – 115.57s	627/628/5176 – 43.66s	18223/18225/159618 – 164.38s

3. Compare and contrast heuristic search result metrics using A* with the "ignore preconditions" and "level-sum" heuristics for Problems 1, 2, and 3.

	S9(A* w/ ignore) Exp/ GoalTest/ NewNode	S10(A* w/ levelsum) Exp/ GoalTest/ NewNode
P1	41/43/170 – 0.042s	11/13/50 – 0.70s
P2	1450/1452/13303 – 4.77s	86/88/841 – 63.09s
P3	5040/5042/44944 – 19.25s	325/327/3002 – 322.48s

4. 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 ignore precondition heuristics performs better in speed, while level-sum heuristics saves much space complexity. Generally, the ignore-precondition heuristics outperforms most of the non-heuristics search in our cases. Besides, Depth First Search also archived a great balance among time performance and space performance although the output plan might not be an optimal one.