Hueristic analysis Report:

Hardware used for testing:
OS: Mc OS Sierra (10.12.6)
CPU: 2.2 GHz Intel Core i7
Memory capacity: 16GB

Test Results:

Problem	Algorithm	Optimal?	Expansion	Goal Tests	New Nodes	Final Plan length	Time elapsed (sec)
	Uniformed Search algorithms						
1	BFS	Yes	43	56	180	6	0.0309
2	BFS	Yes	3343	4609	30509	9	13.632
3	BFS	Yes	14663	18098	129631	12	112.168
1	BFTS	Yes	1458	1459	5960	6	0.8858
2	BFTS		-	-	-	-	-
3	BFTS		-	-	-	-	-
1	DFGS	Yes	21	22	84	20	0.013
2	DFGS	No	624	625	5602	619	3.700
3	DFGS	No	408	409	3364	392	1.88
1	DLS	Yes	101	271	414	50	0.092
2	DLS		-	-	-	-	-
3	DLS		-	-	-	-	-
1	UCS	Yes	55	57	224	6	0.034
2	UCS	Yes	4852	4854	44030	9	11.445
3	UCS	Yes	18235	18237	159716	12	49.91
1	RBFS		4229	4230	17023	6	2.617
2	RBFS		-	-	-	-	-
3	RBFS		-	-	-	-	-
	Heuristic algorithms						
1	GBFS with h_1	Yes	7	9	28	6	0.005

Problem	Algorithm	Optimal?	Expansion	Goal Tests	New Nodes	Final Plan length	Time elapsed (sec)
2	GBFS with h_1	No	990	992	8910	17	4.396
3	GBFS with h_1	No	5614	5616	49429	22	25.63
1	A* WIT H_1	Yes	55	57	224	6	0.034
2	A* WIT H_1	Yes	4852	4854	44030	9	22.03
3	A* WIT H_1	Yes	18235	18237	159716	12	51.63
1	A* with H-IP	Yes	41	43	170	6	0.04
2	A* with H-IP	Yes	1450	1452	13303	9	4.185
3	A* with H-IP	Yes	5040	5042	44944	12	16.95

Optimal Solution for problems

Problem 1 solution	Problem 2 solution	Problem 3 solution
`Load(C1, P1, SFO)`	`Load(C1, P1, SFO)`	`Load(C1, P1, SFO)`
`Load(C2, P2, JFK)`	`Load(C2, P2, JFK)`	`Load(C2, P2, JFK)`
`Fly(P1, SFO, JFK)`	`Load(C3, P3, ATL)`	`Fly(P1, SFO, ATL)`
`Fly(P2, JFK, SFO)`	`Fly(P1, SFO, JFK)`	`Load(C3, P1, ATL)`
`Unload(C1, P1, JFK)`	`Fly(P2, JFK, SFO)`	`Fly(P2, JFK, ORD)`
`Unload(C2, P2, SFO)`	`Fly(P3, ATL, SFO)`	`Load(C4, P2, ORD)`
	`Unload(C3, P3, SFO)`	`Fly(P2, ORD, SFO)`
	`Unload(C1, P1, JFK)`	`Fly(P1, ATL, JFK)`
	`Unload(C2, P2, SFO)`	`Unload(C4, P2, SFO)`
		`Unload(C3, P1, JFK)`
		`Unload(C1, P1, JFK)`
		`Unload(C2, P2, SFO)`

Analysis:
The first problem had a simple search space to look for. As a result all algorithms work well. Depth First Graph Search particularly worked well. However, this algorithm did not provide an optimal solution. So this algorithm can be a very good fit for cases where the search space is very small.

At first glance. GBFS and UCS seemed to have performed well in terms of time taken. However the solution provided is far from optimal. GBFS for the third problem gave this as a result

1. Load(C1, P1, SFO)
2. Load(C2, P2, JFK)
3. Fly(P1, SFO, ORD)
4. Load(C4, P1, ORD)
5. Fly(P2, JFK, ATL)
6. Load(C3, P2, ATL)
7. Fly(P2, ATL, ORD)
8. Fly(P1, ORD, ATL)
9. Unload(C4, P1, ATL)
10.Fly(P1, ATL, ORD)
11.Fly(P2, ORD, ATL)
12.Load(C4, P2, ATL)
13.Fly(P2, ATL, ORD)
14.Unload(C3, P2, ORD)
15.Load(C3, P1, ORD)
16.Fly(P1, ORD, JFK)
17.Unload(C3, P1, JFK)
18.Unload(C1, P1, JFK)
19.Fly(P1, JFK, ORD)
20.Fly(P2, ORD, SFO)
21.Unload(C4, P2, SFO)
22.Unload(C2, P2, SFO)

Unformed Cost Search and Breadth first search are the only algorithms that gave optimal results. But BFS consumed a lot of time before the solution was arrived at. as a result among the uniform search algorithms, UCS is the best solution to go with.

Among the heuristics driven algorithms, A* with h_ignore_preconditions gave the best optimal solution. Best because the number of nodes visited, the time taken and the memory consumed are very less.