Project 3 - Implement a Planning Search

I. Optimal plan for Air Cargo Problems 1, 2, and 3.

a. Air Cargo Problem 1

```
Init(At(C1, SFO) ^ At(C2, JFK) ^ At(P1, SFO) ^ At(P2, JFK)
          ^ Cargo(C1) ^ Cargo(C2)
          ^ Plane(P1) ^ Plane(P2)
          ^ Airport(JFK) ^ Airport(SFO))
Goal(At(C1, JFK) ^ At(C2, SFO))
```

Optimal plan length for problem 1 is = 6

Here are the 6 actions:

1. Load(C1, P1, SFO)
2. Load(C2, P2, JFK)
3. Fly(P1, SFO, JFK)
4. Fly(P2, JFK, SFO)
5. Unload(C1, P1, JFK)
6. Unload(C2, P2, SFO)

b. Air Cargo Problem 2

```
Init(At(C1, SFO) ^ At(C2, JFK) ^ At(C3, ATL)
     ^ At(P1, SFO) ^ At(P2, JFK) ^ At(P3, ATL)
       ^ Cargo(C1) ^ Cargo(C2) ^ Cargo(C3)
       ^ Plane(P1) ^ Plane(P2) ^ Plane(P3)
       ^ Airport(JFK) ^ Airport(SFO) ^ Airport(ATL))
Goal(At(C1, JFK) ^ At(C2, SFO) ^ At(C3, SFO))
```

Optimal plan length for problem 2 is = 9

Here are the 9 actions:

1. Load(C1, P1, SFO)
2. Load(C2, P2, JFK)
3. Load(C3, P3, ATL)
4. Fly(P1, SFO, JFK)

5. Fly(P2, JFK, SFO)
6. Fly(P3, ATL, SFO)
7. Unload(C2, P2, SFO)
8. Unload(C1, P1, JFK)
9. Unload(C3, P3, SFO)

c. Air Cargo Problem 3

```
Init(At(C1, SFO) ^ At(C2, JFK) ^ At(C3, ATL) ^ At(C4, ORD)
        ^ At(P1, SFO) ^ At(P2, JFK)
        ^ Cargo(C1) ^ Cargo(C2) ^ Cargo(C3) ^ Cargo(C4)
        ^ Plane(P1) ^ Plane(P2)
        ^ Airport(JFK) ^ Airport(SFO) ^ Airport(ATL) ^ Airport(ORD))
Goal(At(C1, JFK) ^ At(C3, JFK) ^ At(C2, SFO) ^ At(C4, SFO))
```

Optimal plan length for problem 3 is= **12**

Here are the 12 actions:

1. Load(C1, P1, SFO)
2. Load(C2, P2, JFK)
3. Load(C3, P1, ATL)
4. Load(C4, P2, ORD)
5. Fly(P1, SFO, ATL)
6. Fly(P2, JFK, ORD)
7. Fly(P2, ORD, SFO)
8. Fly(P1, ATL, JFK)
9. Unload(C4, P2, SFO)
10. Unload(C3, P1, JFK)
11. Unload(C2, P2, SFO)
12. Unload(C1, P1, JFK)

II Uninformed Non-heuristic search result metrics

Air Cargo Problem 1						
Search algo	Node Expansions	Goal Tests	New nodes	Time elapsed(seconds)	Plan length	Optimal Plan Len? Y/N
Breadth First Search	43	56	180	0.033	6	Y
Depth First Graph Search	21	22	84	0.015	20	N
greedy_best_first_graph_search h_1	**7**	**9**	**28**	**0.006**	**6**	**Y**
uniform_cost_search	55	57	224	0.036	6	Y

Air Cargo Problem 2						
Search algo	Node Expansions	Goal Tests	New nodes	Time elapsed(seconds)	Plan length	Optimal Plan Len? Y/N
Breadth First Search	3343	4609	30509	14.18	**9**	Y
Depth First Graph Search	**624**	625	5602	**3.18**	619	N
greedy_best_first_graph_search h_1	998	1000	8982	6.59	13	N
uniform_cost_search	4853	4855	44041	49.024	9	Y

Air Cargo Problem 3						
Search algo	Node Expansions	Goal Tests	New nodes	Time elapsed(seconds)	Plan length	Optimal Plan Len? Y/N
Breadth First Search	**14663**	**18098**	**129631**	**100.793**	**12**	**Y**
Depth First Graph Search	**408**	409	3364	**1.63**	392	N
greedy_best_first_graph_search h_1	**5578**	5580	49150	116.48	22	N
uniform_cost_search	18223	18225	159618	417.16	12	Y

Breadth First search always results in an optimal plan length for all three problems. However it is not computationally efficient because it expands more nodes and takes a lot longer to complete. Depth First search in all three problems expands fewer nodes, uses less memory and is significantly faster as seen in problem #3. Greedy best first graph search is a good option as well. While it is not as fast as Depth FirstGraph Search it still has fewer node expansions than Breadth First Search.

III Informed heuristic search result metrics

Air Cargo Problem 1						
Search algo	Node Expansions	Goal Tests	New nodes	Time elapsed(seconds)	Plan length	Optimal Plan Len? Y/N
Breadth First Search	43	56	180	0.033	6	Y
Depth First Graph Search	21	22	84	0.015	20	N
greedy_best_first_graph_search h_1	7	9	28	0.006	6	Y
uniform_cost_search	55	57	224	0.036	6	Y
astar_search h_1	55	57	224	0.039	6	Y
astar_search h_ignore_preconditions	**41**	**43**	**170**	**0.034**	**6**	**Y**
astar_search h_pg_levelsum	11	13	50	1.44	6	Y

Air Cargo Problem 2						
Search algo	Node Expansions	Goal Tests	New nodes	Time elapsed(seconds)	Plan length	Optimal Plan Len? Y/N
Breadth First Search	3343	4609	30509	14.18	9	Y
Depth First Graph Search	624	625	5602	3.18	619	N
greedy_best_first_graph_search h_1	998	1000	8982	6.59	13	N
uniform_cost_search	4853	4855	44041	49.024	9	Y
astar_search h_1	4853	4855	44041	45.946	9	Y

astar_search h_ignore_preconditions	**1506**	**1508**	**13820**	**13.3897**	**9**	**Y**
astar_search h_pg_levelsum	86	88	839	155.179	9	Y

Air Cargo Problem 3						
Search algo	Node Expansions	Goal Tests	New nodes	Time elapsed(seconds)	Plan length	Optimal Plan Len? Y/N
Breadth First Search	14663	18098	129631	100.793	12	Y
Depth First Graph Search	408	409	3364	1.71	392	N
greedy_best_first_graph_search h_1	5578	5580	49150	116.48	22	N
uniform_cost_search	18223	18225	159618	417.16	12	Y
astar_search h_1	18223	18225	159618	385.649	12	Y
astar_search h_ignore_preconditions	**5118**	**5120**	**45650**	**87.03**	**12**	**Y**
astar_search h_pg_levelsum	404	406	3718	1068.75	12	Y

All three A* searches reach the optimal plan length 6, 9, and 12 for problems 1, 2 and 3 respectively.
A* search that ignores preconditions does best for all three air cargo problems. It is the fastest but it also expands more nodes.
A* search with level sum heuristic uses the least memory expanding fewer nodes, but performs poorly compared to search that ignores preconditions.

A* search that ignores preconditons performs better than Breadth First Search in that it reaches optimal plan length faster and expands fewer nodes.