Planning Heuristic Analysis

The 3 planning problems are

- Problem 1 initial state and goal:

```
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))
```

- Problem 2 initial state and goal:

```
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))
```

- Problem 3 initial state and goal:

```
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))
```

<u>Provide an optimal plan for Problems 1, 2, and 3.</u>

Based on my analysis and tests, the optimal plan lengths for problem 1, 2 & 3 are 6,9 and 12 actions respectively. Below are my plans with optimal length:

Problem 1:

- Load(C1, P1, SFO)
- Load(C2, P2, JFK)
- Fly(P2, JFK, SFO)
- Unload(C2, P2, SFO)
- Fly(P1, SFO, JFK)
- Unload(C1, P1, JFK)

Problem 2:

- Load(C1, P1, SFO)
- Load(C2, P2, JFK)

- Load(C3, P3, ATL)
- Fly(P2, JFK, SFO)
- Unload(C2, P2, SFO)
- Fly(P1, SFO, JFK)
- Unload(C1, P1, JFK)
- Fly(P3, ATL, SFO)
- Unload(C3, P3, SFO)

Problem 3:

- Load(C1, P1, SFO)
- Load(C2, P2, JFK)
- Fly(P2, JFK, ORD)
- Load(C4, P2, ORD)
- Fly(P1, SFO, ATL)
- Load(C3, P1, ATL)
- Fly(P1, ATL, JFK)
- Unload(C1, P1, JFK)
- Unload(C3, P1, JFK)
- Fly(P2, ORD, SFO)
- Unload(C2, P2, SFO)
- Unload(C4, P2, SFO)

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; Your third choice of non-heuristic search may be skipped for Problem 3 if it takes longer than 10 minutes to run, but a note in this case should be included.

I tried running all the non-heuristic searches for the problems 1,2 & 3. For problems 2 & 3, breadth first tree search, depth limited search & recursive best first search took a very long time and I cancelled their executions half-way. Below are the results -

Problem 1 results:

Search Type	Optimal	Time Elapsed	Node expansions	Plan length
breadth_first_search	Yes	0.1067	43	6
breadth_first_tree_search	Yes	2.9	1458	6
depth_first_graph_search	No	.041	21	20
depth_limited_search	No	.274	101	50
uniform_cost_search	Yes	.108	55	6
recursive_best_first_search	Yes	8.43	4229	6
greedy_best_first_graph_search	Yes	.0156	7	6

Problem 2 results:

Search Type	Optimal	Time Elapsed	Node expansions	Plan length
breadth_first_search	Yes	64.52	3343	9
breadth_first_tree_search	-	-	-	-
depth_first_graph_search	No	9.16	624	619
depth_limited_search	-	-	-	-
uniform_cost_search	Yes	65.12	4853	9
recursive_best_first_search	-	-	-	-
greedy_best_first_graph_search	No	13.31	998	11

Problem 3 results:

Search Type	Optimal	Time Elapsed	Node expansions	Plan length
breadth_first_search	Yes	416.71	14663	12
breadth_first_tree_search	-	-	-	-
depth_first_graph_search	No	8.95	408	392
depth_limited_search	-	-	-	-
uniform_cost_search	Yes	243.2	18223	12
recursive_best_first_search	-	-	-	-
greedy_best_first_graph_search	No	82.425	5579	22

From the results above, below are my observations

- I see that 'Breadth first search' & 'Uniform cost search' are the only ones that produce an optimal solution under 10 minutes.
- 'Depth first graph search' is the fastest among all other searches(except for problem 1) and does the least number of node expansions but doesn't produce the optimal solution.
- 'Greedy best first search' seems like a good alternative because it has a low run time(compared to others) and finds a near-optimal solution.

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

Problem 1 results:

Search Type	Optimal	Time Elapsed	Node expansions	Plan length
A* search with h1 heuristic	Yes	.245	55	6
A* search with Ignore preconditions heuristic	Yes	.197	41	6

A* search with Level Sum heuristic	Yes	5.29	11	6

Problem 2 results:

Search Type	Optimal	Time Elapsed	Node expansions	Plan length
A* search with h1 heuristic	Yes	68.5	4853	9
A* search with Ignore preconditions heuristic	Yes	23.768	1450	9
A* search with Level Sum heuristic	Yes	850.31	86	9

Problem 3 results:

Search Type	Optimal	Time Elapsed	Node expansions	Plan length
A* search with h1 heuristic	Yes	300.38	18223	12
A* search with Ignore preconditions heuristic	Yes	88.2	5040	12
A* search with Level Sum heuristic	Yes	4637.78	317	12

From the results above, following are my observations –

- All the heuristics generated optimal plans
- A* heuristics with Ignore preconditions is the fastest among the 3 heuristics used above

<u>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?</u>

“A* search with Ignore preconditions heuristic” is the best heuristic overall as it is optimal and takes the least time to compute among other optimal solutions. Also, the results clearly show that the informed search strategies are better than the non-heuristic search planning methods as they are always optimal and generally take less time than the uninformed searches.