Solving Air Cargo problems using uninformed search methods.

I explored 3 uninformed search methods: Breadth First, Depth First and Uniform Cost Search.
Following are the results of test runs for 3 problems:

Air Cargo Problem 1

Search method	Time	Expansions	Goal Tests	New Nodes	Plan length
breadth_first_search	0.0277855	43	56	180	6
depth_first_graph_search	0.01220614	21	22	84	6
uniform_cost_search	0.030791	55	57	224	6

Air Cargo Problem 2

Search method	Time	Expansions	Goal Tests	New Nodes	Plan length
breadth_first_search	10.061483	3343	4609	30509	9
depth_first_graph_search	4.2584862	624	625	5602	9
uniform_cost_search	14.244175	4853	4855	44041	9

Air Cargo Problem 3

Search method	Time	Expansions	Goal Tests	New Nodes	Plan length
breadth_first_search	53.21847586	14663	18098	129631	12
depth_first_graph_search	2.22066	408	409	3364	12
uniform_cost_search	62.149463	18223	18225	159618	12

Considering the above tables, it turns out that Depth First Search works best for the current instances of the problem, but it doesn’t guarantee optimality as it stops whenever it finds the goal. Depth First Search always expands the deepest node in the current frontier of the search tree (AIMA, 3.4). For the current problem description where there seems to be a goal in the left subtree, the depth first search explores the minimum number of nodes and is able to find the goal faster. It event seems to be optimal. Breadth first search and uniform cost search are building the large frontiers with many nodes, and if the solution lies deep enough, it’ll make it harder to reach there. The memory requirements are a bigger problem for both BFS and Uniform Cost Search than the execution time (AIMA, 3.4). However, it’s guaranteed to get the optimal solution there. We can see the proof from our experiments.
In general, exponential-complexity search problems cannot be solved by uninformed methods for any but the smallest instances (AIMA, 3.4).

Solving Air Cargo problems using A* with h_ignore_preconditions and h_level_sum heuristics.

The best heuristic for all 3 problems was the ignore precoditions heuristics (see the tables below for proof). It appears to be that the computational cost for ignore precoditions is O(n), but for levelsum the performance seems to be much slower due to the following: each time the method is called, it’s required to build the PlanningGraph data structure which has the complexity of at least O(n^2). It’s guaranteed to receive the optimal solution using A* search if we have the consistent heuristics (AIMA, 3.5).

Air Cargo Problem 1

Heuristics	Time	Expansions	Goal Tests	New Nodes	Plan length
h_ignore_preconditions	0.0394931	41	43	170	6
h_pg_levelsum	0.7867919	44	46	178	6

Air Cargo Problem 2

Heuristics	Time	Expansions	Goal Tests	New Nodes	Plan length
h_ignore_preconditions	4.81047	1450	1452	13303	9
h_pg_levelsum	569.7005	4853	4855	44041	9

Air Cargo Problem 3

Heuristics	Time	Expansions	Goal Tests	New Nodes	Plan length
h_ignore_preconditions	18.225881	5040	5042	44944	12
h_pg_levelsum	3450.699	18266	18268	160136	12

Considering the above result, the most optimal heuristics so far is h_ignore_preconditions.
Both algorithms returned the optimal plans of the same contents.

Optimal plans for Air Cargo Problems.
Air Cargo Problem 1:
Plan length: 6
Load(C1, P1, SFO)
Load(C2, P2, JFK)
Fly(P1, SFO, JFK)
Unload(C1, P1, JFK)
Fly(P2, JFK, SFO)
Unload(C2, P2, SFO)

Air Cargo Problem 2:
Plan length: 9
Load(C3, P3, ATL)
Fly(P3, ATL, SFO)

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

Air Cargo Problem 3:
Plan length: 12
Load(C2, P2, JFK)
Fly(P2, JFK, ORD)
Load(C4, P2, ORD)
Fly(P2, ORD, SFO)
Unload(C4, P2, SFO)
Load(C1, P1, SFO)
Fly(P1, SFO, ATL)
Load(C3, P1, ATL)
Fly(P1, ATL, JFK)
Unload(C3, P1, JFK)
Unload(C1, P1, JFK)
Unload(C2, P2, SFO)
```