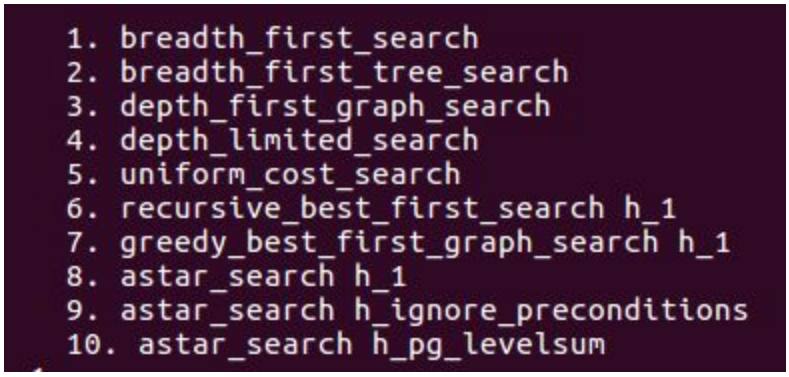
Project analysis report



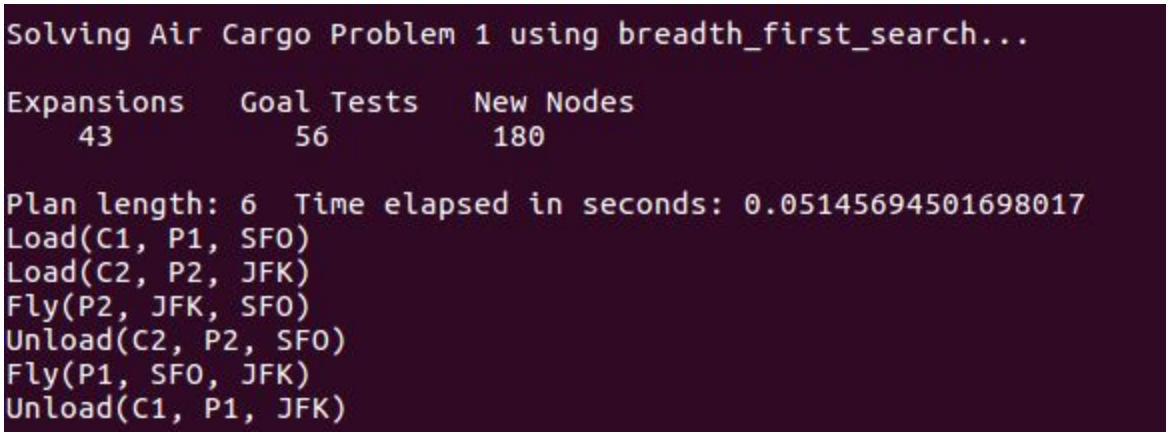
Part 1. Uninformed searches:

Problem 1
For problem 1, it is easy to see that the optimal plan must have at least **6 actions**: load each cargo to the corresponding plane, fly that plane to the other airport and unload it:

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

Below we show the results given by different uninformed search methods (without using any heuristic).

Breadth First Search:



The breadth first search gives the optimal solution. This is expected because all step costs are equal. The graph levels are the same as the number of actions taken and breadth first search give a solution corresponding to the shallowest level node.

Best First Tree Search

This implementation of BFS ignored the revisited state as compared to graph search, so it is expected to be slower and explored more nodes than necessary. However it should still find the optimal solution:

```
Solving Air Cargo Problem 1 using breadth_first_tree_search...

Expansions   Goal Tests   New Nodes
   1458         1459         5960

Plan length: 6  Time elapsed in seconds: 1.2860926389694214
Load(C1, P1, SFO)
Load(C2, P2, JFK)
Fly(P2, JFK, SFO)
Unload(C2, P2, SFO)
Fly(P1, SFO, JFK)
Unload(C1, P1, JFK)
```

Uniform Cost Search

```
Solving Air Cargo Problem 1 using uniform_cost_search...

Expansions   Goal Tests   New Nodes
    55           57          224

Plan length: 6  Time elapsed in seconds: 0.061219099909067154
Load(C1, P1, SFO)
Load(C2, P2, JFK)
Fly(P1, SFO, JFK)
Fly(P2, JFK, SFO)
Unload(C1, P1, JFK)
Unload(C2, P2, SFO)
```

In our problem uniform cost search is very similar to breadth first search since all step cost are equal. However the uniform cost search did not assume anything about the cost structure and must check for all the nodes on the frontier until it can be sure that the founded path is optimal. This is why we have difference in the number of expansions, goal tests …

Depth First Graph Search

```
Solving Air Cargo Problem 1 using depth_first_graph_search...

Expansions   Goal Tests   New Nodes
    21          22          84

Plan length: 20  Time elapsed in seconds: 0.026927973027341068
Fly(P1, SFO, JFK)
Fly(P2, JFK, SFO)
Load(C2, P1, JFK)
Fly(P1, JFK, SFO)
Fly(P2, SFO, JFK)
Unload(C2, P1, SFO)
Fly(P1, SFO, JFK)
Fly(P2, JFK, SFO)
Load(C2, P2, SFO)
Fly(P1, JFK, SFO)
Load(C1, P2, SFO)
Fly(P2, SFO, JFK)
Fly(P1, SFO, JFK)
Unload(C2, P2, JFK)
Unload(C1, P2, JFK)
Fly(P2, JFK, SFO)
Load(C2, P1, JFK)
Fly(P1, JFK, SFO)
Fly(P2, SFO, JFK)
Unload(C2, P1, SFO)
```

Depth first search did not return the optimal solution, which is not surprising. While depth first search is optimal on space complexity (it only need to store only a single path from the root to a leaf node), it returns a solution as soon as it find it without exploring the other branches earlier.

Search Algo	Plan length	Time(s)	Expansions	Goal Tests	New nodes
Breadth First Search	6	0.05	43	56	180
Breadth First Tree Search	6	1.3	1458	1459	5960
Uniform Cost Search	6	0.06	55	57	224
Depth First Search	20	0.03	21	22	84

Depth Limited Search	50	0.14	101	271	414
Recursive BFS with h_1	6	3.8	4229	4230	17023
Greedy BFS with h_1	6	0.01	7	9	28

Problem 2:

Optimal Plan (9 actions):

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

Best First Search

```
Solving Air Cargo Problem 2 using breadth_first_search...

Expansions   Goal Tests   New Nodes
   3343        4609       30509

Plan length: 9  Time elapsed in seconds: 16.72807928500697
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)
```

This takes much longer than the problem 1, but found the optimal solution as expected.

We will not report the detailed results for the other algorithms but only necessary metrics (plan length, time, expansions,...) to compare:

Search Algo	Plan length	Time(s)	Expansions	Goal Tests	New nodes

Breadth First Search	9	16.7	3343	4609	30509
Breadth First Tree Search	*	> 10mins	*	*	*
Uniform Cost Search	9	15.7	4853	4855	44041
Depth First Search	619	4.1	624	625	5602
Depth Limited Search	50	1265	222719	2053741	2054119
Recursive BFS with h_1	*	> 30mins	*	*	*
Greedy BFS with h_1	17	3.1	998	1000	8982

Problem 3:

Optimal Plan (14 actions):

`Load(C1, P1, SFO), Fly(P1, SFO, ATL), Load(C3, P1, ATL), Fly(P1, ATL, JFK), Unload(C1, P1, JFK), Unload(C3, P1, JFK)`

`Load(C2, P2, JFK), Fly(P2, JFK, ORD), Load(C4, P2, ORD), Fly(P2, ORD, SFO), Unload(C2, P2, SFO), Unload(C4, P2, SFO)`

Search result metric tables:
We only run for Breadth First Search, Uniform Cost Search, Depth First Search, Depth Limited Search and Greedy Breadth First Search since all the others will take infinitely.

Search Algo	Plan length	Time(s)	Expansions	Goal Tests	New nodes
Breadth First	12	128	14663	18098	129631

Search					
Uniform Cost Search	12	67	18151	18153	159038
Depth First Search	392	2.19	408	409	3364
Depth Limited Search	*	>30mins	*	*	*
Greedy BFS with h_1	26	20	5398	5400	47665

PART 2

We use A star search using different heuristic functions.
Here are the results for problem 1. All heuristic functions found the optimal plan.

```
Solving Air Cargo Problem 1 using astar_search with h_1...

Expansions   Goal Tests   New Nodes
    55          57          224

Plan length: 6  Time elapsed in seconds: 0.06442110694479197
Load(C1, P1, SFO)
Load(C2, P2, JFK)
Fly(P1, SFO, JFK)
Fly(P2, JFK, SFO)
Unload(C1, P1, JFK)
Unload(C2, P2, SFO)
```

```
Solving Air Cargo Problem 1 using astar_search with h_ignore_preconditions...

Expansions   Goal Tests   New Nodes
    41          43          170

Plan length: 6  Time elapsed in seconds: 0.05843548593111336
Load(C1, P1, SFO)
Fly(P1, SFO, JFK)
Unload(C1, P1, JFK)
Load(C2, P2, JFK)
Fly(P2, JFK, SFO)
Unload(C2, P2, SFO)
```

```
Solving Air Cargo Problem 1 using astar_search with h_pg_levelsum...

Expansions   Goal Tests   New Nodes
    11          13          50

Plan length: 6  Time elapsed in seconds: 0.6322932839393616
Load(C1, P1, SFO)
Fly(P1, SFO, JFK)
Load(C2, P2, JFK)
Fly(P2, JFK, SFO)
Unload(C1, P1, JFK)
Unload(C2, P2, SFO)
```

Problem 1

Search Algo	Plan length	Time(s)	Expansions	Goal Tests	New nodes
A* with h1	6	0.06	55	57	224
A* with h_ignore_preconditions	6	0.06	41	43	170
A* with h_pg_levelsum	6	0.63	11	13	50

Problem 2

Search Algo	Plan length	Time(s)	Expansions	Goal Tests	New nodes
A* with h1	9	16	4853	4855	44041
A* with h_ignore_preconditions	9	6	1450	1452	13303
A* with h_pg_levelsum	9	54	86	88	841

Problem 3

Search Algo	Plan length	Time(s)	Expansions	Goal Tests	New nodes
A* with h1	12	70	18151	18153	159038

A* with h_ignore_preconditions	12	22	5038	5040	44926
A* with h_pg_levelsum	*	> 30mins	*	*	*

With those results we see that whiel h_pg_level sum requires less nodes to be explored (or better heuristic), the calculation time is much longer than the ingore_precondtions one. So the h_ignore_preconditions heuristic gives the best performance.