1. Introduction

In this AIND assignment I have evaluated **three different uninformed and informed (6 overall)** search algorithms using PDDL (Planning Domain Definition Language) to solve three different air cargo shipping problems. The goal was to find an optimal solution for all problems. This document provides a brief summary and discussion of the achieved results for each problem and each class of search algorithm.

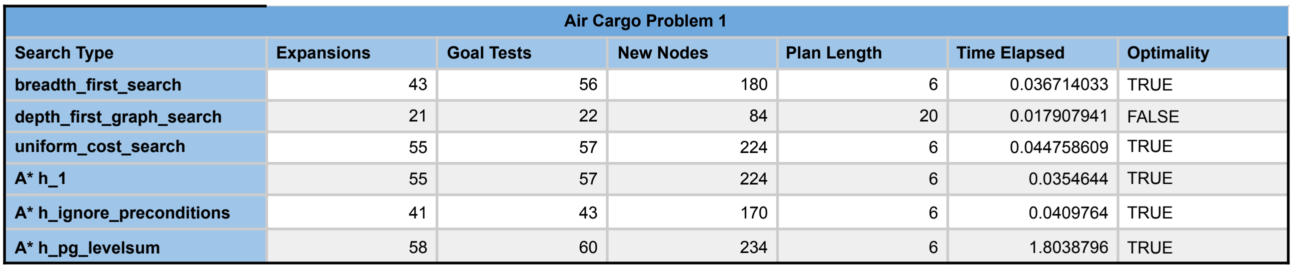
2. Air Cargo Shipping Problems using PDDL

2.1 Problem1

First problem was defined in PDDL as follows:

```
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))
```

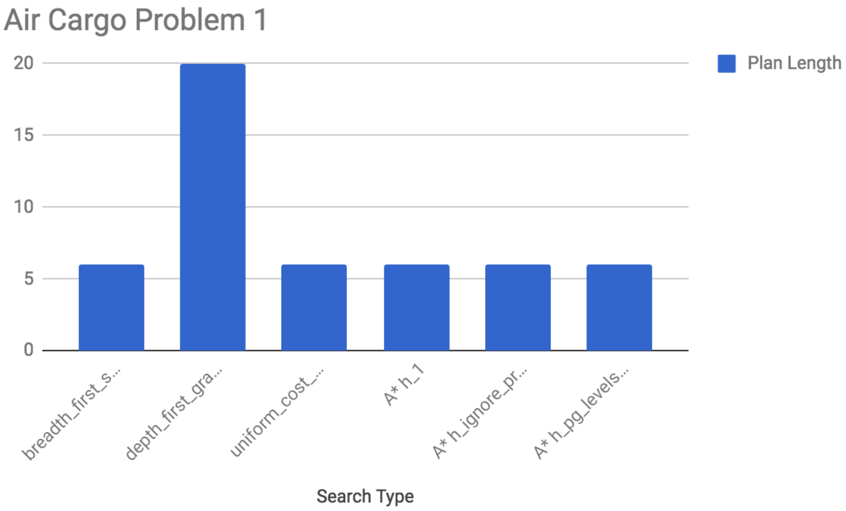
This problem was the easiest of the three, so I expected the fastest results. You can find the breakdown of the results in the table below:



Air Cargo Problem 1						
Search Type	**Expansions**	**Goal Tests**	**New Nodes**	**Plan Length**	**Time Elapsed**	**Optimality**
breadth_first_search	43	56	180	6	0.036714033	TRUE
depth_first_graph_search	21	22	84	20	0.017907941	FALSE
uniform_cost_search	55	57	224	6	0.044758609	TRUE
A* h_1	55	57	224	6	0.0354644	TRUE
A* h_ignore_preconditions	41	43	170	6	0.0409764	TRUE
A* h_pg_levelsum	58	60	234	6	1.8038796	TRUE

You can find the problem 1 program run log in GitHub repo [1] in *p1_uninformed.txt* and *p1_informed.txt* files or see the console screenshots attached in Appendix 1.

Plan length found by each tested algorithm for Problem 1 is illustrated in the chart below:

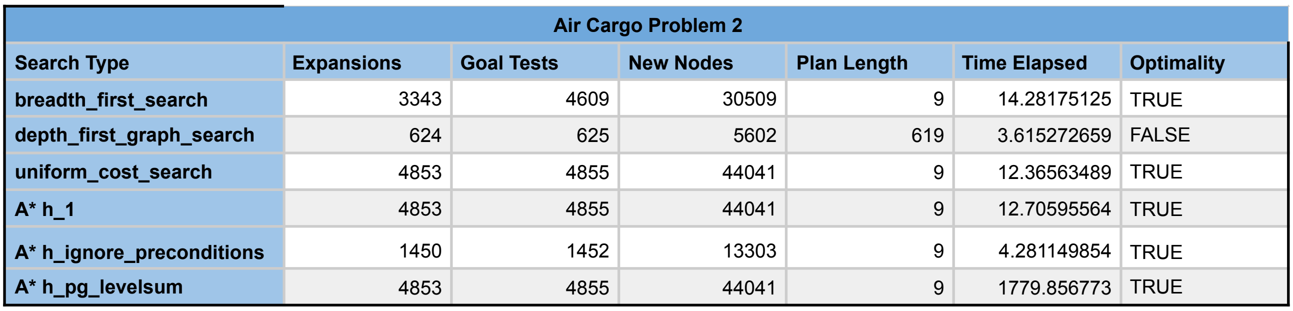


2.2 Problem 2

Second problem was defined in PDDL as follows:

```
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))
```

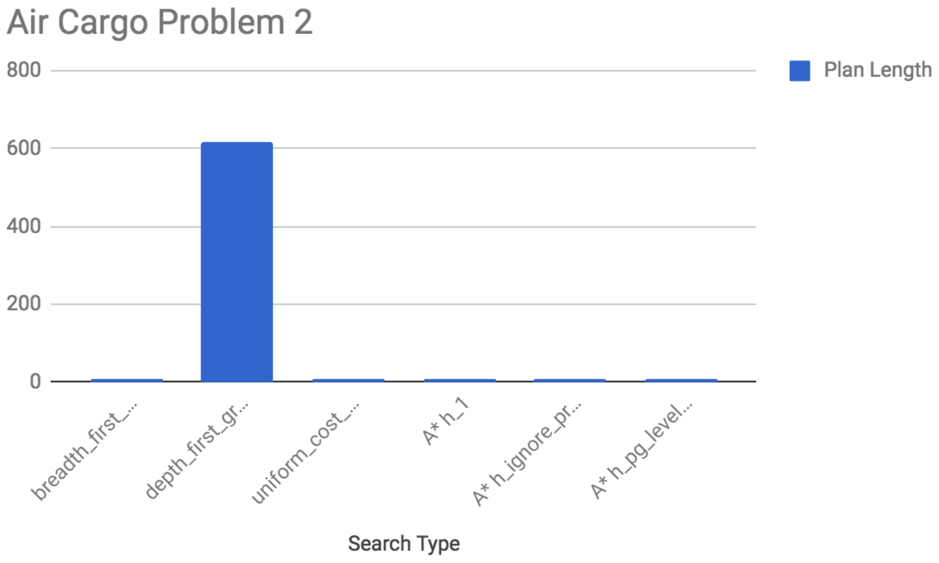
Second problem was visibly slightly more complex. The resulting state space was bigger and the search for the result was expected to take longer than in the case of the first problem. The breakdown of the results confirm this:



Air Cargo Problem 2						
Search Type	**Expansions**	**Goal Tests**	**New Nodes**	**Plan Length**	**Time Elapsed**	**Optimality**
breadth_first_search	3343	4609	30509	9	14.28175125	TRUE
depth_first_graph_search	624	625	5602	619	3.615272659	FALSE
uniform_cost_search	4853	4855	44041	9	12.36563489	TRUE
A* h_1	4853	4855	44041	9	12.70595564	TRUE
A* h_ignore_preconditions	1450	1452	13303	9	4.281149854	TRUE
A* h_pg_levelsum	4853	4855	44041	9	1779.856773	TRUE

You can find the problem 2 program run log in GitHub repo [1] in *p2_uninformed.txt* and *p2_informed.txt* files or see the console screenshots attached in Appendix 1.

Plan length found by each tested algorithm for Problem 2 is illustrated in the chart below:

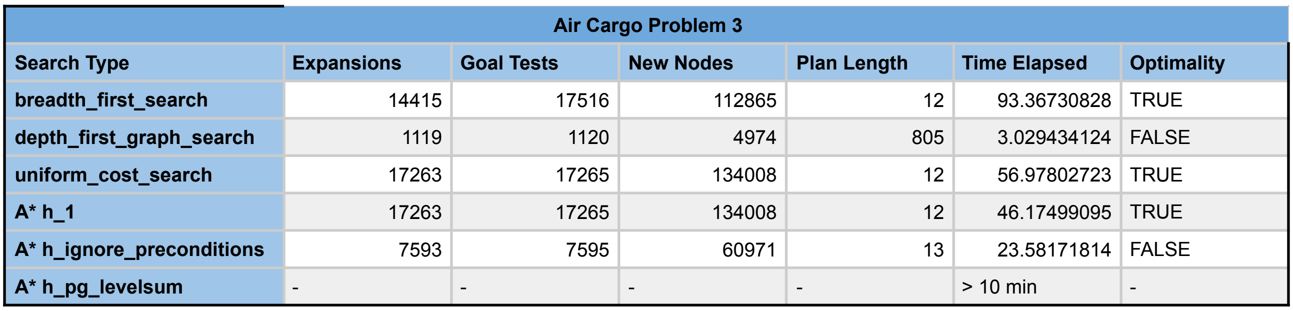


2.3 Problem 3

The third problem was defined in PDDL as follows:

```
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))
```

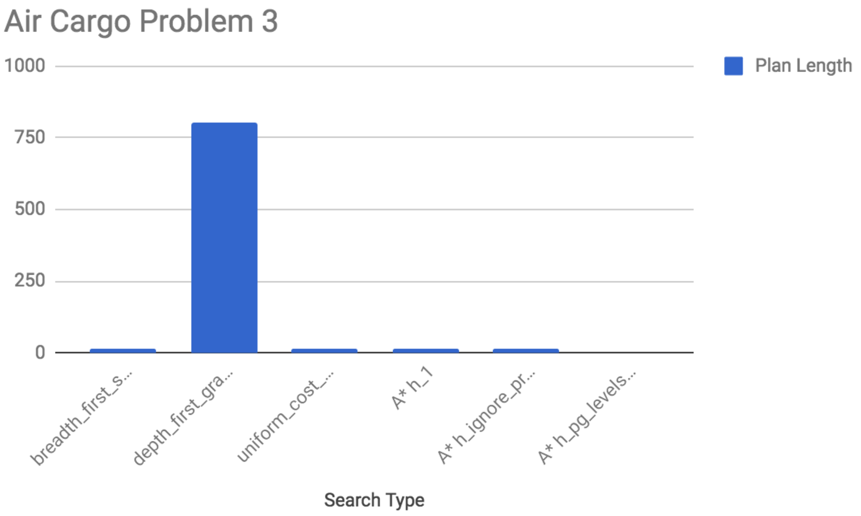
Third problem was visibly the hardest of the three to solve with most constraints and thus biggest state space of the three problems. The breakdown of the results confirm this:



Air Cargo Problem 3						
Search Type	**Expansions**	**Goal Tests**	**New Nodes**	**Plan Length**	**Time Elapsed**	**Optimality**
breadth_first_search	14415	17516	112865	12	93.36730828	TRUE
depth_first_graph_search	1119	1120	4974	805	3.029434124	FALSE
uniform_cost_search	17263	17265	134008	12	56.97802723	TRUE
A* h_1	17263	17265	134008	12	46.17499095	TRUE
A* h_ignore_preconditions	7593	7595	60971	13	23.58171814	FALSE
A* h_pg_levelsum	-	-	-	-	> 10 min	-

You can find the problem 3 program run log in GitHub repo [1] in *p3_uninformed.txt* and *p3_informed.txt* files or see the console screenshots attached in Appendix 1. Please note that *p3_informed.txt* is missing the results of *A* h_pg_levelsum* algorithm as it took a long time to compute and thus was terminated before it finished.

Plan length found by each tested algorithm for Problem 3 is illustrated in the chart below:



3. Experimental results

All the experiments were performed on Google Container Engine (GKE) [2] running in Docker containers using custom built Docker image publicly available on the Docker Hub [3] which packages my source code in it. The source code is also available on Github [1]. The Github repository also contains a Kubernetes job definition [4] which was used to run the *run_search.py* jobs on GKE. All of this allows for an easy experiment reproduction. Furthermore, you can find all the results and chart in the Google spreadsheet [5]. Hardware spec used for these experiments was: 2.4 GHz Intel Core i7, 16 GB 1867 MHz LPDDR3.

3.1 Uninformed search

All uninformed search algorithms find the solution to all three cargo shipping problems.

BFS (Breadth First Search) is guaranteed to find a solution if it exists [6]. Furthermore BFS finds optimal solution [6]. This is nicely demonstrated in the provided results where we can see that BFS finds the optimal result for each cargo shipping problem. However, BFS time complexity is exponential in depth, so the bigger the state space the longer it takes to find a solution, alas the solution will always be optimal if it exists. The time complexity of BFS can be observed in the BFS run for the third, most complex problem.

DFS (Depth First Search) finds the "leftmost" solution regardless of depth or cost [6] as it searches the nodes that go as deep as possible in the state space without considering same depth nodes during the search. This has a nice property of low memory footprint as we keep track of only the explored branch, however the solution found by DFS is not optimal. This is nicely demonstrated in the experimental results provided in this report: the paths to solution found by DFS for each of the problems is the longest one of all three uninformed search algorithms.

Finally I have also tested UCS (Uniformed Cost Search). UCS always finds an optimal solution [6]. I would expect the UCS to perform about the same as BFS which becomes a UCS for path costs of 1.

3.2 Informed search

Informed search algorithms performed better for more complex problems where uninformed search algorithms struggle due to a large size of the problem state space.

A h_1* performed about the same as uninformed searches for first two, simpler problems whilst still managing to find the optimal solution for each of them. The most visible improvement of using *A* h_1* in comparison to using uninformed search algorithms is in the third, most complex problem where the *A* h_1* performed over 50% faster than BFS and almost 20% faster than UCS whilst still finding the optimal solution for the problem.

A h_ignore_precondition* improves the search performance of finding the solution for complex problems (2 and 3) further still, up to 3x faster than uninformed search algorithms. However it does seem to fail to find the optimal solution just about for the third, most complex problem, despite being the fastest of all uninformed and informed algorithms.

It's worth noting that *A* h_pg_levelsum* didn't find the solution for the third problem in reasonable amount of time (>15 minutes) whilst it also took it a long time to find the solution for the second problem. This is most likely due to heuristic being rather complex to calculate: note that this heuristic traverses the graph to add the costs of all partial goals based on their presence at particular graph level!

4. Conclusion

From the provided results we can see that of all **uninformed search algorithms** only BFS and UCS found the optimal solution for all three problems. DFS failed to find the optimal solution, although it found a solution faster than BFS and UCS. Furthermore, it appears that the most optimal uninformed search algorithm for Problem1 is BFS, whilst the best uninformed search algorithm for Problem 2 and 3 is UCS. Whilst both BFS and UCS find the optimal plans, UCS runs faster for Problem 2 and 3.

Of all tested **informed search algorithms** we can see that the best informed search algorithm for problem 1 is *A* with h_1* heuristic. As for the problem 2, *A** with *h_ignore_preconditions* heuristic is the best to use. *A* h_ignore_preconditions* heuristic greatly relaxes the problem which thus takes shorter time to compute. It not only finds the optimal solution for problem 2 it also outperforms the remaining two informed search algorithms in terms of duration. As for the final, most complex problem, it turns out *A* h_1* is the optimal algorithm, alas it's worth noting that *A* h_ignore_preconditions* misses the optimal plan just by one action and runs way faster than *A* h_1* so it could also be considered a reasonably optimal algorithm in cases where the time duration is preferred to optimality without sacrificing too much of the optimality.

Best optimal plans for each problem are listed in **APPENDIX A**. Note that the criteria for choosing the best plan was optimality and time duration of the run algorithm. When more than one algorithm finds the optimal goal the faster plans are preferred to the slower ones.

APPENDIX B contains screenshots of each algorithm run.

References:

[1] https://github.com/milosgajdos83/udacity-aind/tree/master/planning-ai
[2] https://cloud.google.com/container-engine/
[3] https://hub.docker.com/r/gyre007/planning/
[4] https://kubernetes.io/docs/concepts/workloads/controllers/jobs-run-to-completion/
[5] https://docs.google.com/spreadsheets/d/1-ybgCV6lF-p4_YDX6F6AVGnWBhU6GA1ZT-3KarLPejY/edit#gid=0
[6] Russell, S. J., Norvig, P. (2010), Artificial intelligence: A modern approach.

APPENDIX A

Best optimal solutions for each problem:

Problem 1, BFS:

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

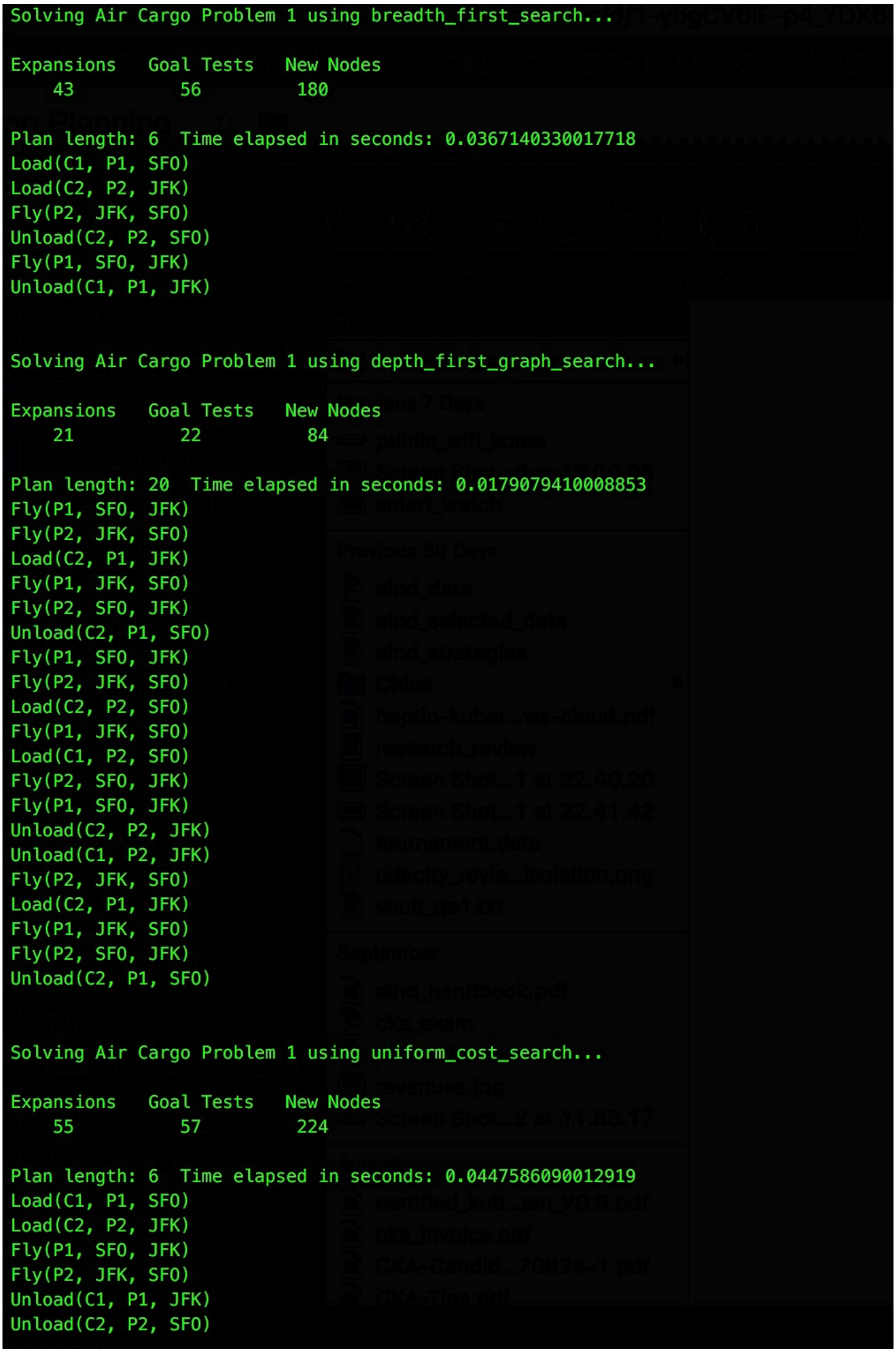
Problem 2, A* h_ignore_preconditions:

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

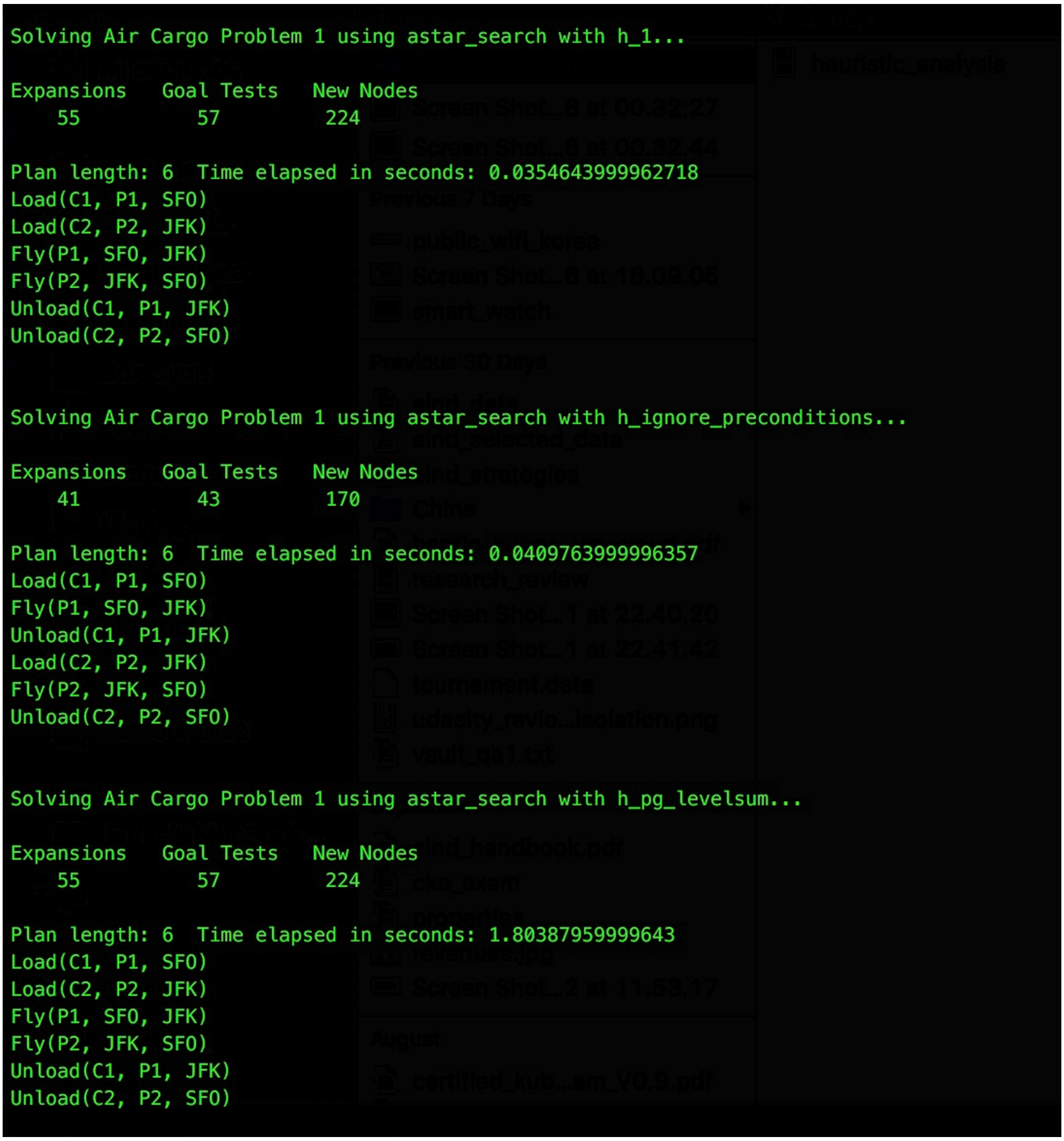
Problem 3, A* h_1:

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

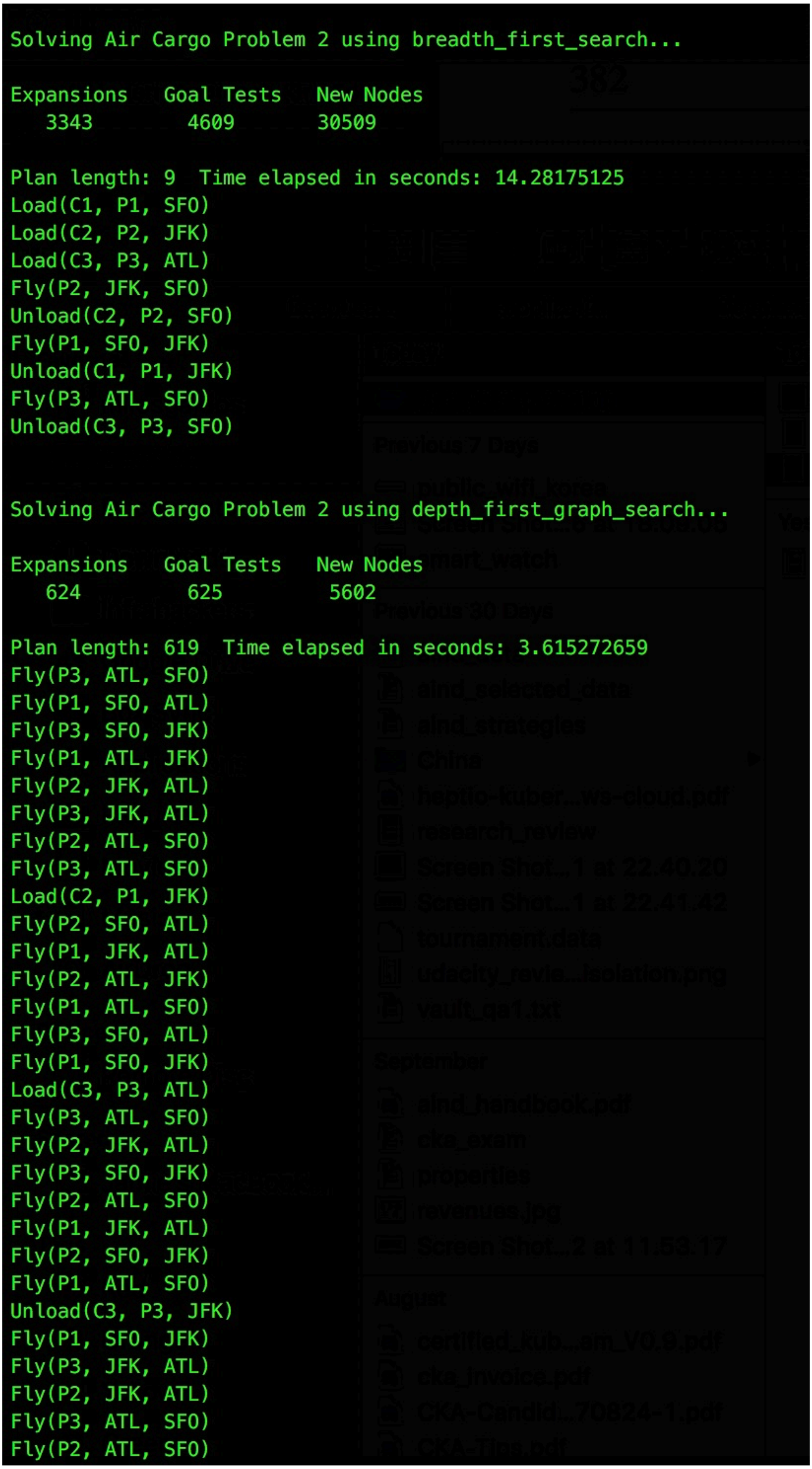
APPENDIX B



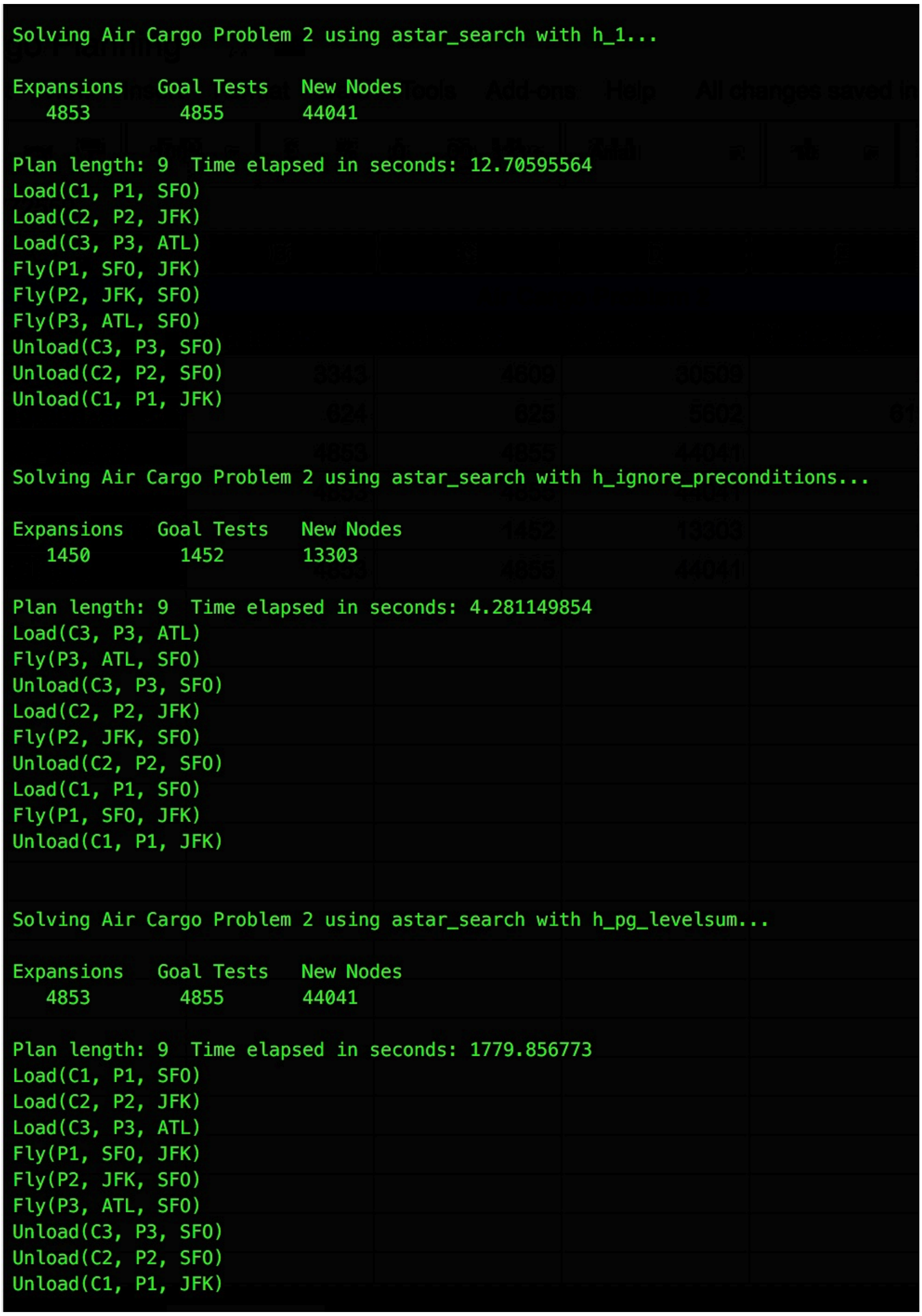
```
Solving Air Cargo Problem 1 using breadth_first_search...

Expansions   Goal Tests   New Nodes
    43           56          180

Plan length: 6  Time elapsed in seconds: 0.0367140330017718
Load(C1, P1, SFO)
Load(C2, P2, JFK)
Fly(P2, JFK, SFO)
Unload(C2, P2, SFO)
Fly(P1, SFO, JFK)
Unload(C1, P1, JFK)


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: 0.0179079410008853
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)


Solving Air Cargo Problem 1 using uniform_cost_search...

Expansions   Goal Tests   New Nodes
    55           57          224

Plan length: 6  Time elapsed in seconds: 0.0447586090012919
Load(C1, P1, SFO)
Load(C2, P2, JFK)
Fly(P1, SFO, JFK)
Fly(P2, JFK, SFO)
Unload(C1, P1, JFK)
Unload(C2, P2, SFO)
```



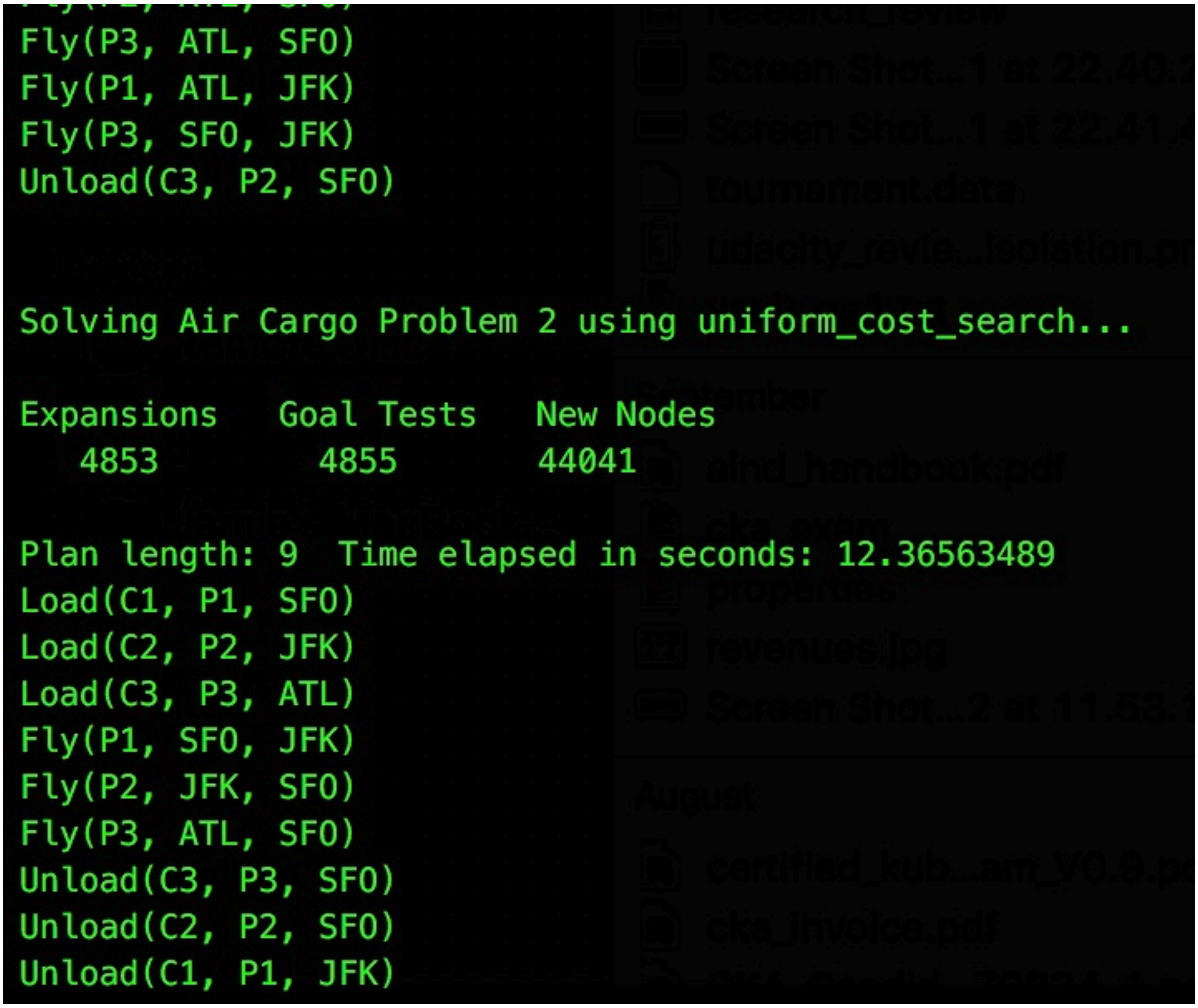
```
Solving Air Cargo Problem 1 using astar_search with h_1...

Expansions   Goal Tests   New Nodes
    55          57          224

Plan length: 6  Time elapsed in seconds: 0.0354643999962718
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: 0.0409763999996357
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
    55          57          224

Plan length: 6  Time elapsed in seconds: 1.80387959999643
Load(C1, P1, SFO)
Load(C2, P2, JFK)
Fly(P1, SFO, JFK)
Fly(P2, JFK, SFO)
Unload(C1, P1, JFK)
Unload(C2, P2, SFO)
```



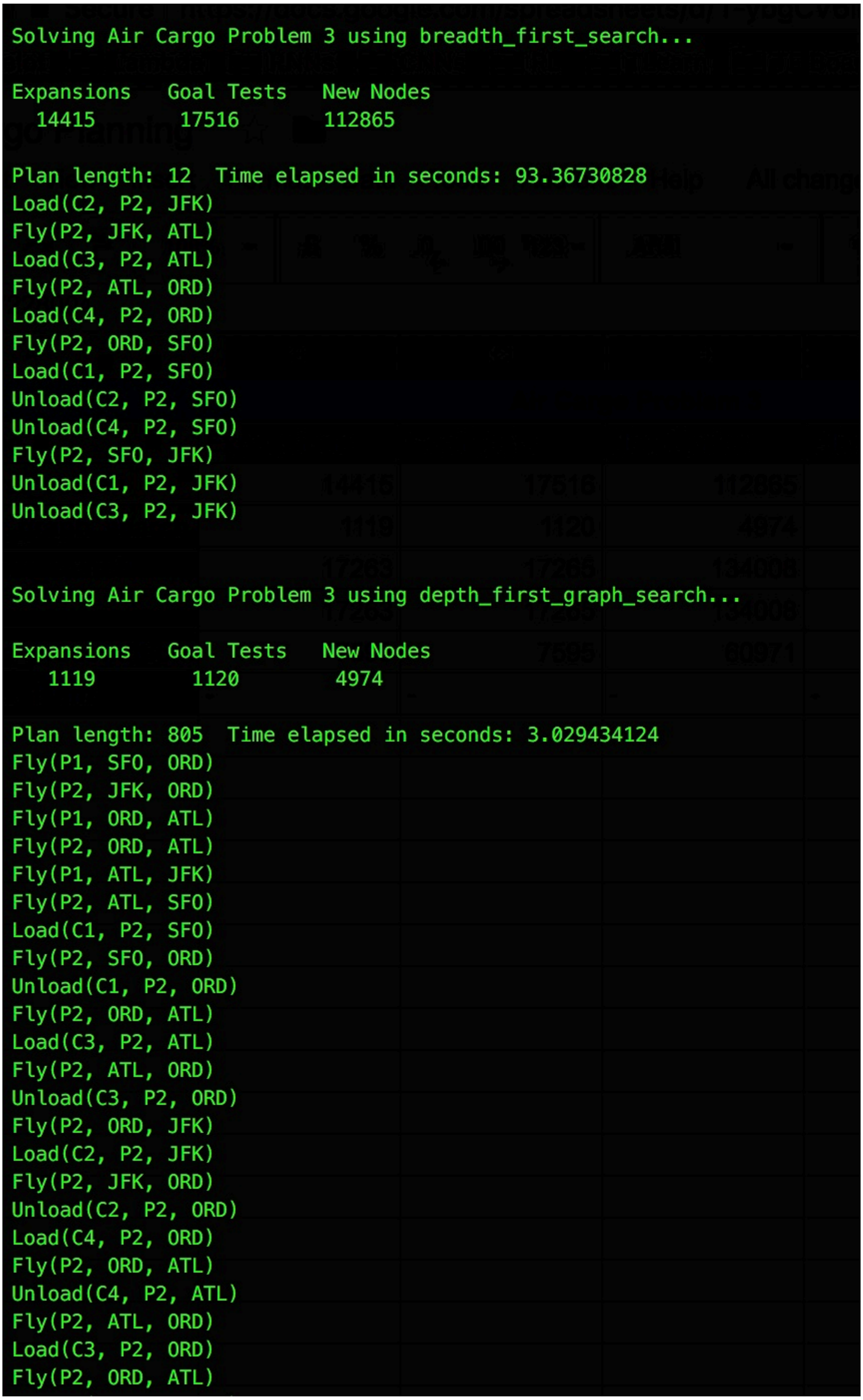
```
Solving Air Cargo Problem 2 using breadth_first_search...

Expansions   Goal Tests   New Nodes
   3343         4609        30509

Plan length: 9  Time elapsed in seconds: 14.28175125
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)


Solving Air Cargo Problem 2 using depth_first_graph_search...

Expansions   Goal Tests   New Nodes
   624          625          5602

Plan length: 619  Time elapsed in seconds: 3.615272659
Fly(P3, ATL, SFO)
Fly(P1, SFO, ATL)
Fly(P3, SFO, JFK)
Fly(P1, ATL, JFK)
Fly(P2, JFK, ATL)
Fly(P3, JFK, ATL)
Fly(P2, ATL, SFO)
Fly(P3, ATL, SFO)
Load(C2, P1, JFK)
Fly(P2, SFO, ATL)
Fly(P1, JFK, ATL)
Fly(P2, ATL, JFK)
Fly(P1, ATL, SFO)
Fly(P3, SFO, ATL)
Fly(P1, SFO, JFK)
Load(C3, P3, ATL)
Fly(P3, ATL, SFO)
Fly(P2, JFK, ATL)
Fly(P3, SFO, JFK)
Fly(P2, ATL, SFO)
Fly(P1, JFK, ATL)
Fly(P2, SFO, JFK)
Fly(P1, ATL, SFO)
Unload(C3, P3, JFK)
Fly(P1, SFO, JFK)
Fly(P3, JFK, ATL)
Fly(P2, JFK, ATL)
Fly(P3, ATL, SFO)
Fly(P2, ATL, SFO)
```



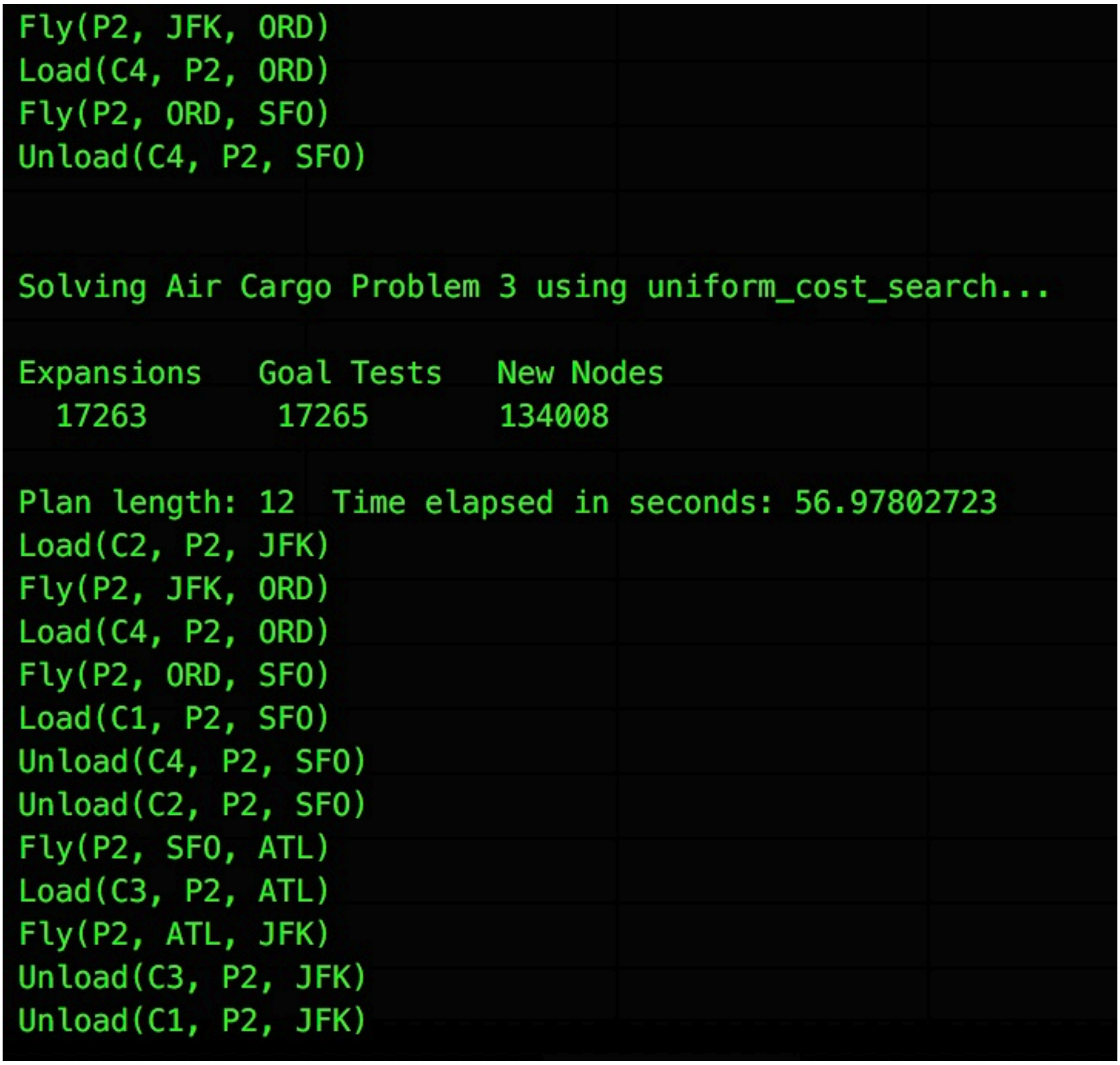
```
Solving Air Cargo Problem 2 using astar_search with h_1...

Expansions   Goal Tests   New Nodes
   4853         4855       44041

Plan length: 9  Time elapsed in seconds: 12.70595564
Load(C1, P1, SFO)
Load(C2, P2, JFK)
Load(C3, P3, ATL)
Fly(P1, SFO, JFK)
Fly(P2, JFK, SFO)
Fly(P3, ATL, SFO)
Unload(C3, P3, SFO)
Unload(C2, P2, SFO)
Unload(C1, P1, JFK)


Solving Air Cargo Problem 2 using astar_search with h_ignore_preconditions...

Expansions   Goal Tests   New Nodes
   1450         1452       13303

Plan length: 9  Time elapsed in seconds: 4.281149854
Load(C3, P3, ATL)
Fly(P3, ATL, SFO)
Unload(C3, P3, SFO)
Load(C2, P2, JFK)
Fly(P2, JFK, SFO)
Unload(C2, P2, SFO)
Load(C1, P1, SFO)
Fly(P1, SFO, JFK)
Unload(C1, P1, JFK)


Solving Air Cargo Problem 2 using astar_search with h_pg_levelsum...

Expansions   Goal Tests   New Nodes
   4853         4855       44041

Plan length: 9  Time elapsed in seconds: 1779.856773
Load(C1, P1, SFO)
Load(C2, P2, JFK)
Load(C3, P3, ATL)
Fly(P1, SFO, JFK)
Fly(P2, JFK, SFO)
Fly(P3, ATL, SFO)
Unload(C3, P3, SFO)
Unload(C2, P2, SFO)
Unload(C1, P1, JFK)
```



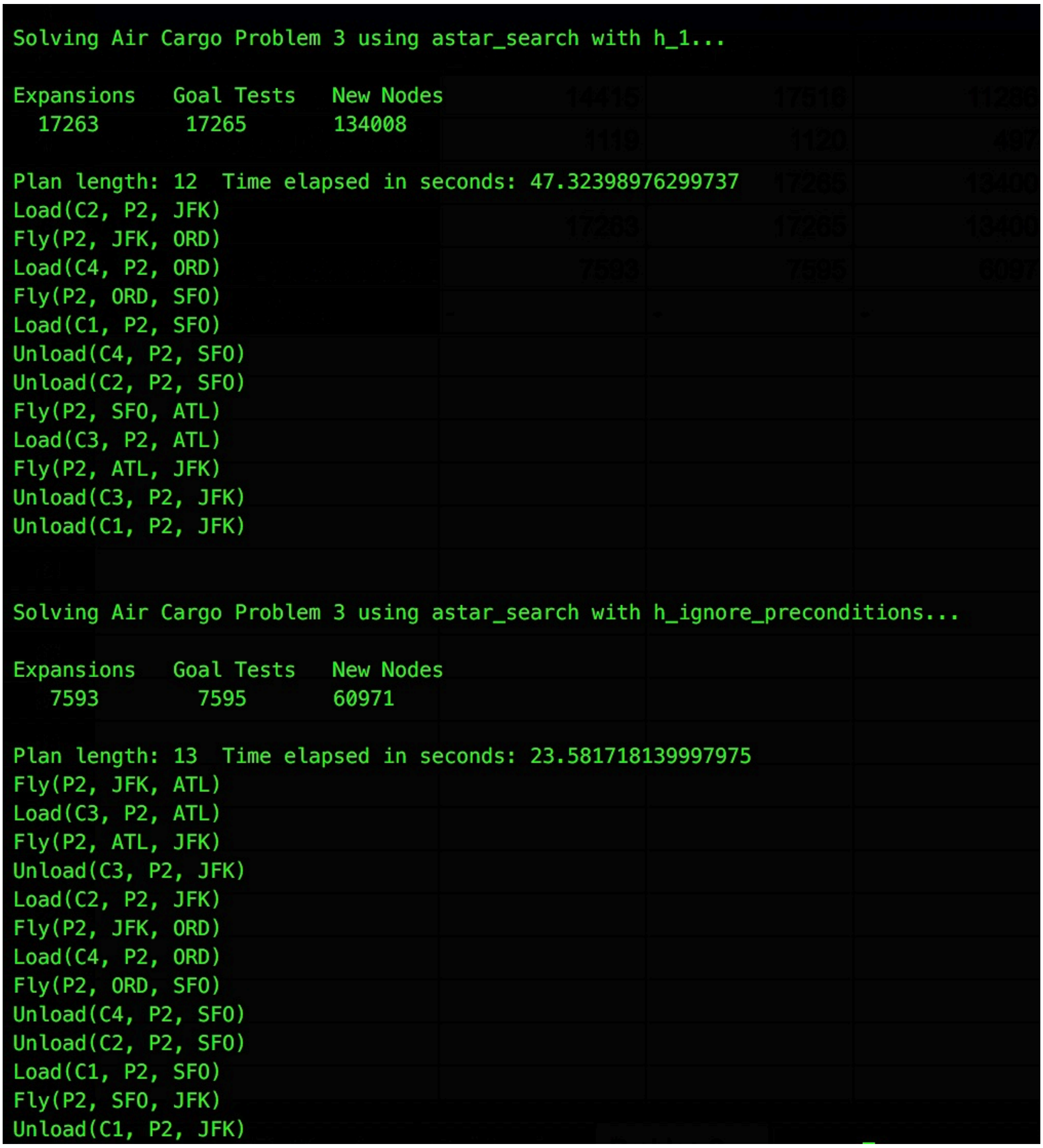
```
Fly(P3, ATL, SFO)
Fly(P1, ATL, JFK)
Fly(P3, SFO, JFK)
Unload(C3, P2, SFO)


Solving Air Cargo Problem 2 using uniform_cost_search...

Expansions   Goal Tests   New Nodes
   4853         4855        44041

Plan length: 9  Time elapsed in seconds: 12.36563489
Load(C1, P1, SFO)
Load(C2, P2, JFK)
Load(C3, P3, ATL)
Fly(P1, SFO, JFK)
Fly(P2, JFK, SFO)
Fly(P3, ATL, SFO)
Unload(C3, P3, SFO)
Unload(C2, P2, SFO)
Unload(C1, P1, JFK)
```



```
Solving Air Cargo Problem 3 using breadth_first_search...

Expansions   Goal Tests   New Nodes
  14415        17516       112865

Plan length: 12  Time elapsed in seconds: 93.36730828
Load(C2, P2, JFK)
Fly(P2, JFK, ATL)
Load(C3, P2, ATL)
Fly(P2, ATL, ORD)
Load(C4, P2, ORD)
Fly(P2, ORD, SFO)
Load(C1, P2, SFO)
Unload(C2, P2, SFO)
Unload(C4, P2, SFO)
Fly(P2, SFO, JFK)
Unload(C1, P2, JFK)
Unload(C3, P2, JFK)


Solving Air Cargo Problem 3 using depth_first_graph_search...

Expansions   Goal Tests   New Nodes
   1119         1120         4974

Plan length: 805  Time elapsed in seconds: 3.029434124
Fly(P1, SFO, ORD)
Fly(P2, JFK, ORD)
Fly(P1, ORD, ATL)
Fly(P2, ORD, ATL)
Fly(P1, ATL, JFK)
Fly(P2, ATL, SFO)
Load(C1, P2, SFO)
Fly(P2, SFO, ORD)
Unload(C1, P2, ORD)
Fly(P2, ORD, ATL)
Load(C3, P2, ATL)
Fly(P2, ATL, ORD)
Unload(C3, P2, ORD)
Fly(P2, ORD, JFK)
Load(C2, P2, JFK)
Fly(P2, JFK, ORD)
Unload(C2, P2, ORD)
Load(C4, P2, ORD)
Fly(P2, ORD, ATL)
Unload(C4, P2, ATL)
Fly(P2, ATL, ORD)
Load(C3, P2, ORD)
Fly(P2, ORD, ATL)
```



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


Solving Air Cargo Problem 3 using uniform_cost_search...

Expansions   Goal Tests   New Nodes
  17263        17265        134008

Plan length: 12  Time elapsed in seconds: 56.97802723
Load(C2, P2, JFK)
Fly(P2, JFK, ORD)
Load(C4, P2, ORD)
Fly(P2, ORD, SFO)
Load(C1, P2, SFO)
Unload(C4, P2, SFO)
Unload(C2, P2, SFO)
Fly(P2, SFO, ATL)
Load(C3, P2, ATL)
Fly(P2, ATL, JFK)
Unload(C3, P2, JFK)
Unload(C1, P2, JFK)
```



```
Solving Air Cargo Problem 3 using astar_search with h_1...

Expansions   Goal Tests   New Nodes
  17263        17265      134008

Plan length: 12  Time elapsed in seconds: 47.32398976299737
Load(C2, P2, JFK)
Fly(P2, JFK, ORD)
Load(C4, P2, ORD)
Fly(P2, ORD, SFO)
Load(C1, P2, SFO)
Unload(C4, P2, SFO)
Unload(C2, P2, SFO)
Fly(P2, SFO, ATL)
Load(C3, P2, ATL)
Fly(P2, ATL, JFK)
Unload(C3, P2, JFK)
Unload(C1, P2, JFK)


Solving Air Cargo Problem 3 using astar_search with h_ignore_preconditions...

Expansions   Goal Tests   New Nodes
   7593         7595       60971

Plan length: 13  Time elapsed in seconds: 23.581718139997975
Fly(P2, JFK, ATL)
Load(C3, P2, ATL)
Fly(P2, ATL, JFK)
Unload(C3, P2, JFK)
Load(C2, P2, JFK)
Fly(P2, JFK, ORD)
Load(C4, P2, ORD)
Fly(P2, ORD, SFO)
Unload(C4, P2, SFO)
Unload(C2, P2, SFO)
Load(C1, P2, SFO)
Fly(P2, SFO, JFK)
Unload(C1, P2, JFK)
```