Heuristic Analysis - Air Cargo Search

Air Cargo Problem 1

Uninformed Algorithm	Plan Length	Expansions	Goal Tests	New Nodes	Time (seconds)
BFS	6	43	56	180	0.0147
BFTS	6	1458	1459	5960	0.452
DFGS	20	21	22	84	0.0068
DLS	50	101	271	414	0.044
UCS	6	55	57	224	0.0187
Heuristic Algorithm					
RBFS w/ h1	6	4229	4230	17023	1.346
GBFGS w/ h1	6	7	9	28	0.0026
A* w/ h1	6	55	57	224	0.0184
A* w/ h_ig_prec	6	41	43	170	0.0188
A* w/ h_pg_levsum	6	11	13	50	0.5620

Air Cargo Problem 2

Uninformed Algorithm	Plan Length	Expansions	Goal Tests	New Nodes	Time (seconds)
BFS	9	3343	4609	30509	6.6892
BFTS	DNR				
DFGS	619	624	625	5602	1.676
DLS	50	222719	2053741	2054119	436.4
UCS	9	4853	4855	44041	5.615
Heuristic Algorithm					
RBFS w/ h1	DNR				
GBFGS w/ h1	15	998	1000	8982	1.156
A* w/ h1	9	4853	4855	44041	5.699
A* w/ h_ig_prec	9	1450	1452	13303	2.141
A* w/ h_pg_levsum	9	86	88	841	87.65

Air Cargo Problem 3

Uninformed Algorithm	Plan Length	Expansions	Goal Tests	New Nodes	Time (seconds)
BFS	12	14663	18098	129631	50.3
BFTS	DNR				
DFGS	392	408	409	3364	0.8633
DLS	DNR				
UCS	12	18223	18225	159618	29.45
Heuristic Algorithm					
RBFS w/ h1	DNR				
GBFGS w/ h1	22	5578	5580	49150	9.212
A* w/ h1	12	18223	18225	159618	24.61
A* w/ h_ig_prec	12	5040	5042	44944	8.002
A* w/ h_pg_levsum	12	325	327	3002	548.8

Result Analysis

For the definition of an optimal solution, we will use the definition from AI: A Modern Approach by Norvig and Russel(1, pg 68).

> "*A solution to a problem is an action sequence that leads from the initial state to a goal state. Solution quality is measured by the OPTIMAL SOLUTION path cost function, and an optimal solution has the* ***lowest path cost among all solutions****."*

As a secondary metric, we will measure the efficiency of the solution by its runtime, as this is fundamentally the measurable cost of running a planning algorithm on processing hardware.

1) *Provide an optimal plan for problems 1, 2, and 3.*

Problem 1, being significantly easier to solve than the others, converges to a somewhat misleading optimal solution of **greedy best first tree search** using heuristic h1 as this happened in this case to find an optimal (minimal length) solution to the problem in the least runtime with minimal nodes expanded. The breadth first search uninformed algorithm also managed to outperform the A* searches in this problem due to it's simplicity, but by a lesser margin. With that being said, our knowledge of the increased optimality of A* search gives some insight into why the A* search algorithms which converged on the optimal path slightly slower will be much more sustainable as the search becomes more difficult.
Optimal: Algorithm 7 - greedy best first tree search w/ heuristic 1

Problem 2, which demands a much more extensive search, exposed the speedy GBFTS for it's lack of optimality, as it converged on a non-optimal solution of length 15, while the optimal search methods were able to find the length 9 solution. In this problem, **A* search using the relaxed heuristic ignoring preconditions** was the optimal solution in that it found the shortest possible plan length of 9 in the shortest time.
Optimal: Algorithm 9 - A* w/ h_ignore_preconditions

Problem 3, the most complex of our planning problems, further shows that **A* search using the relaxed heuristic** is our optimal planning algorithm for the problem, as it was able to find the optimal solution in the least amount of time.
Optimal: Algorithm 9 - A* w/ h_ignore_preconditions

2) 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.

Problem 1

BFS	6	43	56	180	0.0147
DFGS	20	21	22	84	0.0068
UCS	6	55	57	224	0.0187

Problem 2

BFS	9	3343	4609	30509	6.6892
DFGS	619	624	625	5602	1.676
UCS	9	4853	4855	44041	5.615

Problem 3

BFS	12	14663	18098	129631	50.3
DFGS	392	408	409	3364	0.8633
UCS	12	18223	18225	159618	29.45

Breadth-first search in the air cargo domain proved to be a feasible solution to our problem space, however in the real world as the problem space becomes exponentially more complex, the exponential increase in runtime looks to be a significant red flag if this were to be considered as a valuable method. While it is by nature optimal in that it will always find the solution of the shortest length and least cost, it must be noted that this necessity for excessive node expansion and processing power will lead this to be an infeasible solution compared to heuristic search as the problems become more complex. When compared to **depth-first search**, it will inevitably be slower, yet the fact that DFS will rarely stumble upon the optimal solution makes it impractical for situations in which finding the least-cost solution is necessary. While DFS is non-optimal but quite efficient, and BFS is optimal but significantly less efficient, **uniform-cost search** was in fact our best performer using our definition of optimality. UCS was slower for the simpler problem, but as the problem complexity increased its runtime increased much less than BFS, making it the best uninformed search algorithm in that it found the optimal plan in the least amount of time.

BFS: Optimal, somewhat efficient
DFS: Non-Optimal, extremely efficient
UCS: Optimal, more efficient than BFS at scale

3) 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

A* w/ h_ig_prec	6	41	43	170	0.0188
A* w/ h_pg_levsum	6	11	13	50	0.5620

Problem 2

A* w/ h_ig_prec	9	1450	1452	13303	2.141
A* w/ h_pg_levsum	9	86	88	841	87.65

Problem 3

A* w/ h_ig_prec	12	5040	5042	44944	8.002
A* w/ h_pg_levsum	12	325	327	3002	548.8

The comparison here can potentially be misleading, as if one were to approach the efficiency of their solution to merely be based on the number of nodes being expanded and traversed, they would be led to believe that the level sum heuristic is more efficient. However the obvious difference is the amount of processing power required to calculate the heuristic value for the level sum, which at scale becomes incredibly memory intensive and thus each node expanded will take significantly longer to evaluate. The ignoring preconditions heuristic on the other hand is meant to be less intensive than a standard heuristic for the problem and it succeeds in doing so, outperforming both h1 and level sum in terms of processing speed. It is important to note that it does so while maintaining the optimality of A* search, meaning that with this heuristic we will still manage to find the absolute least-cost solution

4) 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?

Problem 1

A* w/ h1	6	55	57	224	0.0184
A* w/ h_ig_prec	6	41	43	170	0.0188
A* w/ h_pg_levsum	6	11	13	50	0.5620

Problem 2

A* w/ h1	9	4853	4855	44041	5.699
A* w/ h_ig_prec	9	1450	1452	13303	2.141
A* w/ h_pg_levsum	9	86	88	841	87.65

Problem 3

A* w/ h1	12	18223	18225	159618	24.61
A* w/ h_ig_prec	12	5040	5042	44944	8.002
A* w/ h_pg_levsum	12	325	327	3002	548.8

While h1 is admissible but not really a true estimation of the distance to the goal and more of a placeholder, it has been included for the sake of comparison. For the reasons described in the previous response, our **heuristic ignoring preconditions** is in fact our best heuristic for this problem in that it will always find the optimal solution, and it will tend to find it in less time than any other heuristic we evaluated. The one exception in this case is in problem 1 where our placeholder heuristic was insignificantly faster than h_ignore_preconditions due to the fact that it requires no processing, however it gets exposed by the exponential growth of the number of nodes expanding while the problem becomes more complex. This same misleading result in problem 1 can lead one to believe that some of the uninformed searches (BFS,DFGS, and UCS) slightly edge out our best heuristic search in terms of efficiency, however it quickly becomes obvious that the rising complexity of the problem makes those searches impractical through lack of optimality and/or efficiency. For that reason, the ignoring preconditions heuristic will always outperform uninformed search methods as well as our level sum heuristic in a realistic (at least somewhat complex) problem space.

Best Algorithm: 9 **- A* search with h_ignoring_preconditions**

Citations

1) SJ Russell, P Norvig, “Artificial intelligence: a modern approach (3rd edition)”, 2007

Evaluation Log

```
>python run_search.py -p 1 -s 1
```

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

Expansions   Goal Tests   New Nodes
    43          56         180

Plan length: 6  Time elapsed in seconds: 0.014682483959472014
Load(C1, P1, SFO)
Load(C2, P2, JFK)
Fly(P2, JFK, SFO)
Unload(C2, P2, SFO)
Fly(P1, SFO, JFK)
Unload(C1, P1, JFK)
```

```
>python run_search.py -p 1 -s 2
```

```
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: 0.45196320940180174
Load(C1, P1, SFO)
Load(C2, P2, JFK)
Fly(P2, JFK, SFO)
Unload(C2, P2, SFO)
Fly(P1, SFO, JFK)
Unload(C1, P1, JFK)
```

```
>python run_search.py -p 1 -s 3
```

```
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: 0.006809123117803365
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)
```

>python run_search.py -p 1 -s 4

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

Expansions   Goal Tests   New Nodes
   101         271         414

Plan length: 50  Time elapsed in seconds: 0.04395752770636652
Load(C1, P1, SFO)
Load(C2, P2, JFK)
Unload(C1, P1, SFO)
Load(C1, P1, SFO)
Unload(C1, P1, SFO)
Load(C1, P1, SFO)
Unload(C1, P1, SFO)
Load(C1, P1, SFO)
Unload(C1, P1, SFO)
Load(C1, P1, SFO)
```

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

```
>python run_search.py -p 1 -s 5
```

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

Expansions   Goal Tests   New Nodes
    55          57         224

Plan length: 6  Time elapsed in seconds: 0.01867992395601547
Load(C1, P1, SFO)
Load(C2, P2, JFK)
Fly(P1, SFO, JFK)
Fly(P2, JFK, SFO)
Unload(C1, P1, JFK)
Unload(C2, P2, SFO)
```

```
>python run_search.py -p 1 -s 6
```

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

Expansions   Goal Tests   New Nodes
   4229        4230       17023

Plan length: 6  Time elapsed in seconds: 1.346004720343926
Load(C2, P2, JFK)
Load(C1, P1, SFO)
Fly(P2, JFK, SFO)
Unload(C2, P2, SFO)
Fly(P1, SFO, JFK)
Unload(C1, P1, JFK)
```

```
>python run_search.py -p 1 -s 7
```

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

Expansions   Goal Tests   New Nodes
    7           9          28

Plan length: 6  Time elapsed in seconds: 0.002561893754455702З
Load(C1, P1, SFO)
Load(C2, P2, JFK)
Fly(P1, SFO, JFK)
Fly(P2, JFK, SFO)
Unload(C1, P1, JFK)
Unload(C2, P2, SFO)
```

```
>python run_search.py -p 1 -s 8

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: 0.018371805396530494
Load(C1, P1, SFO)
Load(C2, P2, JFK)
Fly(P1, SFO, JFK)
Fly(P2, JFK, SFO)
Unload(C1, P1, JFK)
Unload(C2, P2, SFO)
```

```
>python run_search.py -p 1 -s 9

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: 0.018803063362785977
Load(C1, P1, SFO)
Fly(P1, SFO, JFK)
Unload(C1, P1, JFK)
Load(C2, P2, JFK)
Fly(P2, JFK, SFO)
Unload(C2, P2, SFO)
```

```
>python run_search.py -p 1 -s 10

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: 0.5620195942880598
Load(C1, P1, SFO)
Fly(P1, SFO, JFK)
Load(C2, P2, JFK)
Fly(P2, JFK, SFO)
```

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

>python run_search.py -p 2 -s 1

Solving Air Cargo Problem 2 using breadth_first_search...

Expansions   Goal Tests   New Nodes
   3343        4609       30509

Plan length: 9  Time elapsed in seconds: 6.688668474151526
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)

>python run_search.py -p 2 -s 2
DNR

>python run_search.py -p 2 -s 3

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: 1.67638909892199

>python run_search.py -p 2 -s 4

Solving Air Cargo Problem 2 using depth_limited_search...

Expansions   Goal Tests   New Nodes
  222719     2053741     2054119

Plan length: 50  Time elapsed in seconds: 436.4176054246149
Load(C1, P1, SFO)
Load(C2, P2, JFK)
Load(C3, P3, ATL)
Unload(C1, P1, SFO)
Load(C1, P1, SFO)
Unload(C1, P1, SFO)
Load(C1, P1, SFO)
Unload(C1, P1, SFO)
Load(C1, P1, SFO)
Unload(C1, P1, SFO)
Load(C1, P1, SFO)
Unload(C1, P1, SFO)
Load(C1, P1, SFO)
Unload(C1, P1, SFO)
Load(C1, P1, SFO)
Unload(C1, P1, SFO)
Load(C1, P1, SFO)
Unload(C1, P1, SFO)
Load(C1, P1, SFO)
Unload(C1, P1, SFO)
Load(C1, P1, SFO)
Unload(C1, P1, SFO)
Load(C1, P1, SFO)
Unload(C1, P1, SFO)
Load(C1, P1, SFO)
Unload(C1, P1, SFO)
Load(C1, P1, SFO)
Unload(C1, P1, SFO)
Load(C1, P1, SFO)
Unload(C1, P1, SFO)
Load(C1, P1, SFO)
Unload(C1, P1, SFO)
Load(C1, P1, SFO)
Unload(C1, P1, SFO)
Load(C1, P1, SFO)
Unload(C1, P1, SFO)
Load(C1, P1, SFO)
Unload(C1, P1, SFO)
Load(C1, P1, SFO)
Unload(C1, P1, SFO)
Load(C1, P1, SFO)
Unload(C1, P1, SFO)
Load(C1, P1, SFO)
Fly(P2, JFK, SFO)
Unload(C2, P2, SFO)
```

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

```
>python run_search.py -p 2 -s 5

Solving Air Cargo Problem 2 using uniform_cost_search...

Expansions   Goal Tests   New Nodes
   4853        4855       44041

Plan length: 9  Time elapsed in seconds: 5.615407278187042
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)
```

```
>python run_search.py -p 2 -s 6
DNR
```

```
>python run_search.py -p 2 -s 7

Solving Air Cargo Problem 2 using greedy_best_first_graph_search with h_1...

Expansions   Goal Tests   New Nodes
   998         1000       8982

Plan length: 15  Time elapsed in seconds: 1.1563330380868024
Load(C1, P1, SFO)
Load(C2, P2, JFK)
Load(C3, P3, ATL)
Fly(P1, SFO, ATL)
Fly(P2, JFK, ATL)
Fly(P3, ATL, SFO)
Fly(P2, ATL, SFO)
Unload(C2, P2, SFO)
```

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

>python run_search.py -p 2 -s 8

```
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: 5.698695154356327
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)
```

>python run_search.py -p 2 -s 9

```
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: 2.1414698956555553
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)
```

>python run_search.py -p 2 -s 10

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

Expansions   Goal Tests   New Nodes
    86          88         841

Plan length: 9  Time elapsed in seconds: 87.6517663483765
Load(C1, P1, SFO)
Fly(P1, SFO, JFK)
Load(C2, P2, JFK)
Fly(P2, JFK, SFO)
Load(C3, P3, ATL)
Fly(P3, ATL, SFO)
Unload(C3, P3, SFO)
Unload(C2, P2, SFO)
Unload(C1, P1, JFK)
```

>python run_search.py -p 3 -s 1

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

Expansions   Goal Tests   New Nodes
  14663       18098       129631

Plan length: 12  Time elapsed in seconds: 50.29766494199486
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)
```

>python run_search.py -p 3 -s 2
DNR

```
>python run_search.py -p 3 -s 3

Solving Air Cargo Problem 3 using depth_first_graph_search...

Expansions   Goal Tests   New Nodes
   408        409        3364

Plan length: 392  Time elapsed in seconds: 0.8633058069951824
---
```

```
>python run_search.py -p 3 -s 4
DNR
```

```
>python run_search.py -p 3 -s 5

Solving Air Cargo Problem 3 using uniform_cost_search...

Expansions   Goal Tests   New Nodes
  18223      18225      159618

Plan length: 12  Time elapsed in seconds: 29.450151763917994
Load(C1, P1, SFO)
Load(C2, P2, JFK)
Fly(P1, SFO, ATL)
Load(C3, P1, ATL)
Fly(P2, JFK, ORD)
Load(C4, P2, ORD)
Fly(P2, ORD, SFO)
Fly(P1, ATL, JFK)
Unload(C4, P2, SFO)
Unload(C3, P1, JFK)
Unload(C2, P2, SFO)
Unload(C1, P1, JFK)
```

```
>python run_search.py -p 3 -s 6
DNR
```

```
>python run_search.py -p 3 -s 7

Solving Air Cargo Problem 3 using greedy_best_first_graph_search with h_1...

Expansions   Goal Tests   New Nodes
   5578       5580       49150
```

```
Plan length: 22  Time elapsed in seconds: 9.212038767309728
Load(C1, P1, SFO)
Load(C2, P2, JFK)
Fly(P1, SFO, ORD)
Load(C4, P1, ORD)
Fly(P2, JFK, ATL)
Load(C3, P2, ATL)
Fly(P2, ATL, ORD)
Fly(P1, ORD, ATL)
Unload(C4, P1, ATL)
Fly(P1, ATL, ORD)
Fly(P2, ORD, ATL)
Load(C4, P2, ATL)
Fly(P2, ATL, ORD)
Unload(C3, P2, ORD)
Load(C3, P1, ORD)
Fly(P1, ORD, JFK)
Unload(C3, P1, JFK)
Unload(C1, P1, JFK)
Fly(P1, JFK, ORD)
Fly(P2, ORD, SFO)
Unload(C4, P2, SFO)
Unload(C2, P2, SFO)
```

>python run_search.py -p 3 -s 8

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

Expansions   Goal Tests   New Nodes
  18223       18225       159618

Plan length: 12  Time elapsed in seconds: 24.607797208840115
Load(C1, P1, SFO)
Load(C2, P2, JFK)
Fly(P1, SFO, ATL)
Load(C3, P1, ATL)
Fly(P2, JFK, ORD)
Load(C4, P2, ORD)
Fly(P2, ORD, SFO)
Fly(P1, ATL, JFK)
Unload(C4, P2, SFO)
Unload(C3, P1, JFK)
Unload(C2, P2, SFO)
Unload(C1, P1, JFK)
```

>python run_search.py -p 3 -s 9

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

Expansions   Goal Tests   New Nodes
   5040        5042       44944

Plan length: 12  Time elapsed in seconds: 8.001732052971548
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(C2, P2, SFO)
Unload(C1, P1, JFK)
```

>python run_search.py -p 3 -s 10

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

Expansions   Goal Tests   New Nodes
   325         327         3002

Plan length: 12  Time elapsed in seconds: 548.8512702801962
Load(C2, P2, JFK)
Fly(P2, JFK, ORD)
Load(C4, P2, ORD)
Fly(P2, ORD, SFO)
Load(C1, P1, SFO)
Fly(P1, SFO, ATL)
Load(C3, P1, ATL)
Fly(P1, ATL, JFK)
Unload(C4, P2, SFO)
Unload(C3, P1, JFK)
Unload(C2, P2, SFO)
```

Unload(C1, P1, JFK)