Provide an optimal plan for Problems 1, 2, and 3.

Problem 1

```
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 2

```
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)
```

Problem 3

```
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)
```

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

Search	Expansions	Goal Tests	New Nodes	Plan length	Time elapsed
1.breadth-first	43	56	180	6	0.05485508700076025
3.depth-first	21	22	84	12	0.026748197997221723
5.uninformed-cost	55	57	224	6	0.06200596899725497

Problem 2

Search	Expansions	Goal Tests	New Nodes	Plan length	Time elapsed
1.breadth-first	3343	4609	30509	9	13.735633240998 4
3.depth-first	624	625	5602	619	3.445357781994 9714
5.uninformed-co st	4852	4854	44030	9	12.536586457994 417

Problem 3

Search	Expansions	Goal Tests	New Nodes	Plan length	Time elapsed
1.breadth-first	14663	18098	129631	12	104.7384856580029
3.depth-first	408	409	3364	392	1.823144302994478 5
5.uninformed-co st	18235	18237	159716	12	55.4538823790062 45

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

Search	Expansions	Goal Tests	New Nodes	Plan length	Time elapsed
astar_search h_1	55	57	224	6	0.063888392993249
astar_search h_ignore_preconditions	41	43	170	6	0.064539627986960 11
astar_search h_pg_levelsum	11	13	50	6	0.558340440009487 8

Problem 2

Search	Expansions	Goal Tests	New Nodes	Plan length	Time elapsed
astar_search h_1	4852	4854	44030	9	12.879883941001026
astar_search h_ignore_preconditions	1450	1452	13303	9	4.938015168000886
astar_search h_pg_levelsum	86	88	841	9	45.054660214984324

Problem 3

Search	Expansions	Goal Tests	New Nodes	Plan length	Time elapsed
astar_search h_1	18235	18237	159716	12	54.506712503003655
astar_search h_ignore_preconditions	5040	5042	44944	12	17.717213969997833
astar_search h_pg_levelsum	325	327	3002	12	234.63691881799605

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?

- Why does DFS not provide an optimal plan for this case?
  - DFS' purpose is just find one possible solution.
  - DFS has no guarantee find optimal solution.
- Which are the fastest and slowest uninformed algorithms? Why?[1]
  - Fastest uninformed algorithm : DFS
    - DFS find just one solution. So, DFS check very small nodes compare to BFS.
  - Slowest uninformed algorithm : BFS
    - BFS check all possible nodes until find solution. So It must check large combinations.
- Which are the fastest and slowest heuristic searches? Why?[2]
  - Fastest heuristic search : h_ignore_preconditions

[1] Udacity Artificial Intelligence Nanodegree - Lesson 8: Search - Search Comparison
[2] Udacity Artificial Intelligence Nanodegree - Lesson 8: Optimistic Heuristic

        - It's very simple heuristic function.
    - Slowest heuristic search : h_pg_levelsum
        - Complicated heuristic function need very long time for calculate each heuristic result.
- Which algorithms (uninformed and heuristic) have the lowest node expansions and goal tests? Why? What does this mean for the performance in this problem?
    - Lowest node expansions : astar_search h_pg_levelsum
    - Lowest goal tests : astar_search h_pg_levelsum
        - H_pg_levelsum heuristic function need very small number of node expansions and goal tests.
        - So, if node expansion and goal test's cost is very high compared to heuristic function execution time, h_pg_levelsum heuristic function can be good choice.
        - But in this test, node expansion and goal test is very low. So, h_pg_levelsum's heuristic function execution time is high compared to node expansion and goal test.
- If need optimal solution : exclude depth-first.
    - If time elapsed is most important : choose a* with h_ignore_preconditions.
    - If spatial efficiency is most important : choose astar_search h_pg_levelsum.
- If optimal is not important and Time elapsed is most important : depth-first.
    - So non-heuristic search(depth-first) can be chosen.
    - It's very fast.