Part 1: Metrics for non-heuristic planning solution searches

I ran `breadth_first_search`, `depth_first_graph_search`, and `uniform_cost_search`. Here are the results:

	BFS	DFS	Uniform Cost
Problem 1	Length: 6 Time: 0.03s Expansions: 43 Goal Tests: 56 New Nodes: 180	Length: 20 Time: 0.02s Expansions: 21 Goal Tests: 22 New Nodes: 84	Length: 6 Time: 0.05s Expansions: 55 Goal Tests: 57 New Nodes: 224
Problem 2	Length: 9 Time: 12.6s Expansions: 3343 Goal Tests: 4609 New Nodes: 30509	Length: 619 Time: 3.0s Expansions: 624 Goal Tests: 625 New Nodes: 5602	Length: 9 Time: 10.9s Expansions: 4852 Goal Tests: 4854 New Nodes: 44030
Problem 3	Length: 12 Time: 89.8s Expansions: 14663 Goal Tests: 18098 New Nodes: 129631	Length: 392 Time: 1.5s Expansions: 408 Goal Tests: 409 New Nodes: 3364	Length: 12 Time: 45.6s Expansions: 18236 Goal Tests: 18238 New Nodes: 159726

Part 2: Metrics for A* searches with heuristics

	A *Ignore Preconditions Heuristic*	**A Level Sum Heuristic**
	Length: 6	Length: 6

Problem 1	Time: 0.05s Expansions: 41 Goal Tests: 43 New Nodes: 170	Time: 0.55s Expansions: 11 Goal Tests: 13 New Nodes: 50
Problem 2	Length: 9 Time: 3.6s Expansions: 1450 Goal Tests: 1452 New Nodes: 13303	Length: 9 Time: 41.7s Expansions: 86 Goal Tests: 88 New Nodes: 841
Problem 3	Length: 12 Time: 14.8s Expansions: 5040 Goal Tests: 5042 New Nodes: 44944	Length: 12 Time: 245.1s Expansions: 318 Goal Tests: 320 New Nodes: 2934

Part 3: Written Analysis

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

Optimal plan for Problem 1:

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

Optimal plan for 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(C1, P1, JFK)
Unload(C2, P2, SFO)
```

Optimal plan for 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(C3, P1, JFK)
Unload(C4, P2, SFO)
Unload(C1, P1, JFK)
Unload(C2, P2, SFO)
```

Compare and contrast non-heuristic search result metrics (optimality, time elapsed, number of node expansions) for Problems 1,2, and 3.

- **BFS:**
    - **Optimality:** Finds an optimal solution, because of the nature of BFS - if it finds a solution, then it must be the shortest path, given that the cost of a path is the same. (from the Udacity video lectures on search)
    - **Time Elapsed:** Much slower than DFS. Probably because the branching factor is large.
    - **Number of Node Expansions:** Much larger than DFS - BFS is less memory effecient than DFS because it must keep all the nodes on a level in the memory (from the Udacity video lectures on search)
- **DFS:**
    - **Optimality:** Does not find an optimal solution, because of the nature of DFS - if a solution exists at both a shallower level and a deeper level, DFS might find a deeper level first. (from the Udacity video lectures on search)
    - **Time Elapsed:** Much faster than BFS. Probably because the branching factor is large and searching deep is faster than searching wide.
    - **Number of Node Expansions:** Much smaller than BFS because DFS is more memory efficient. (from the Udacity video lectures on search)
- **Uniform Cost:**
    - **Optimality:** Finds an optimal solution, because this is pretty much equivalent to BFS.
    - **Time Elapsed:** Slightly faster than BFS because it uses memoization.
    - **Number of Node Expansions:** Slightly more than BFS, probably because it uses memoization.

Compare and contrast heuristic search result metrics using A* with the "ignore preconditions" and "level-sum" heuristics for Problems 1, 2, and 3.

- **A* Ignore Preconditions Heuristic:**
    - **Optimality:** Finds an optimal solution, because A* with an admissible heuristic is guaranteed to find the shortest path. (from the Udacity video lectures on search)
    - **Time Elapsed:** Very fast, almost as fast as DFS, because computing the heuristic is cheap.
    - **Number of Node Expansions:** Much more compared to level sum.

- **A* Level Sum Heuristic:**
  - **Optimality:** Finds an optimal solution, because A* with an admissible heuristic is guaranteed to find the shortest path.
  - **Time Elapsed:** Not as fast as the ignore precontisions heuristic, slower than BFS, probably because constructing a planning graph is costly time wise.
  - **Number of Node Expansions:** Smallest out of all methods, even more efficient than DFS.

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?

If you care about speed, then ignore preconditions is the best heuristic. But if you care about memory, or if node expansions/goal tests are costly, then level sum heuristic is better.

If you don't care about optimality at all but care about speed, DFS might be the way to go. There's no point in using BFS because it performed more poorly than A* with ignore preconditions in all aspects.