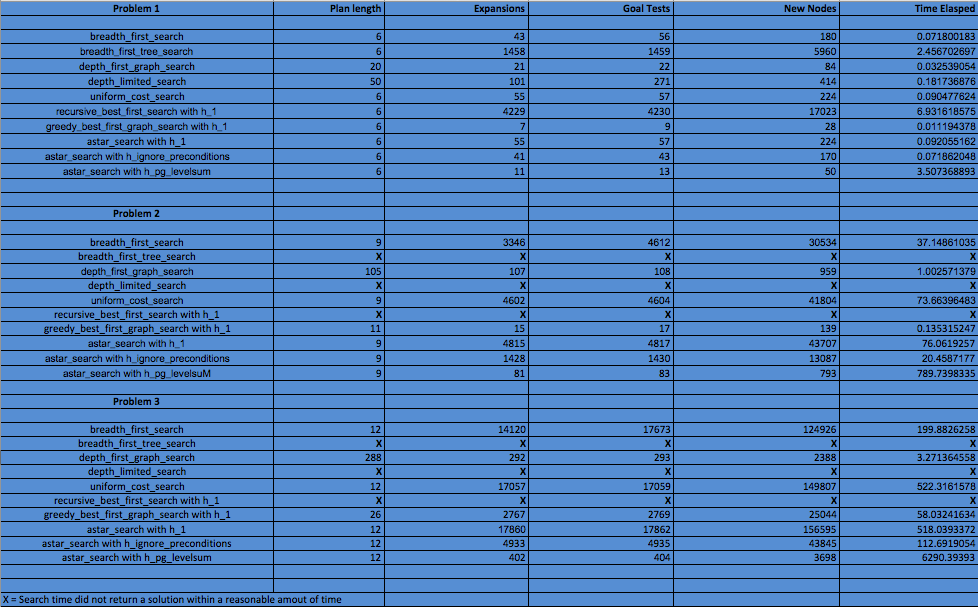
**Planning Search Algorithm**

**Description**:

Three problems are defined in classical PDDL and implemented in my\_air\_cargo\_problems.py. Domain-independent heuristics are defined by implementing relaxed problem heuristics in my\_air\_cargo\_problems.py and implementing Planning Graph and automatic heuristics in my\_planning\_graph.py. Various uninformed and informed planning searches are run to find solutions to each of the planning problems.

**Results:**



**Observations**:

Generally, as the complexity of a planning problem increases (as seen in the result from problem 1 to 3) the metrics increases as well. For breadth first search and uniform cost search, resulting metrics, with the exception of plan length, increase exponentially with increase in the complexity of a planning problem. On the other hand, depth first graph search has a linear increase as complexity increases. Results also show that both breadth first search and uniform cost search find optimal solutions but takes more time to compute while depth first graph search produces a non-optimal solution but takes less time to compute.

Results show that A\* search with heuristics, ignore preconditions and level-sum, produce optimal solutions for all problems. Though A\* search with level-sum search fewer nodes than A\* search with ignore preconditions. A\* search with level-sum takes more time to compute compared to A\* search with ignore preconditions due to additional computation in searching for a solution in a planning graph.

With a tradeoff in computational time, the best performing heuristics using A\* search is level-sum as it produces an optimal solution and searches far fewer nodes than the other heuristics. It also performs better compared to the non-heuristic search planning methods in terms of optimality and nodes expanded.

**Recommended Plans**:

Using A\* search with level-sum, the following are optimal plans for the planning problems.

**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)

Unload(C2, P2, SFO)

Load(C3, P3, ATL)

Fly(P3, ATL, SFO)

Unload(C3, P3, 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(C3, P1, JFK)

Unload(C4, P2, SFO)

Unload(C1, P1, JFK)

Unload(C2, P2, SFO)