Congratulations! Great work!

Planning Graph Implementation

✓ (AUTOGRADED) Student code passes all Project Assistant test cases for:

- `ActionLayer` mutual exclusion rules:
  - `_inconsistent_effects()`
  - `_interference()`
  - `_competing_needs()`
- `LiteralLayer` mutual exclusion rules:
  - `_inconsistent_support()`
  - `_negation()`

Correct! (Note: this rubric item was graded automatically.)

Heuristic Implementation

✓ (AUTOGRADED) Student code passes all Project Assistant test cases for:
Correctly implemented

- `PlanningGraph` class heuristics:
  - `h_levelsum()`
  - `h_maxlevel()`
  - `h_setlevel()`

Correct! (Note: this rubric item was graded automatically.)

Experimental Results & Report

✓ Report includes a table or chart to analyze the number of nodes expanded against number of actions in the domain.

- The chart or table includes data for all search & heuristic combinations for air cargo problems 1 and 2
- The chart or table includes data at least one uninformed search, two heuristics with greedy best first search, and two heuristics with A* on air cargo problems 3 and 4
- Report includes at least a one paragraph discussion of these results that analyzes the growth trends as the problem size increases

👍 Great tables!

Indeed, as the number of action increases complexities (times, expansions, etc) grow in more order of magnitude.

✓ Report includes a table or chart to analyze the search time against the number of actions in the domain.

- The chart or table includes data for all search & heuristic combinations for air cargo problems 1 and 2
- The chart or table includes data at least one uninformed search, two heuristics with greedy best first search, and two heuristics with A* on air cargo problems 3 and 4
- Report includes at least a one paragraph discussion of these results that analyzes the growth trends as the problem size increases

✓ Report includes a table or chart to analyze the length of the plans returned by each algorithm on all search problems.

- The chart or table includes data for all search & heuristic combinations for air cargo problems 1 and 2
- The chart or table includes data at least one uninformed search, two heuristics with greedy best first search, and two heuristics with A* on air cargo problems 3 and 4

✓ Submission includes a short answer to each of the following questions. (A short answer should be at least 1-2 sentences at most a small paragraph.)

- Which algorithm or algorithms would be most appropriate for planning in a very restricted domain (i.e., one that has only a few actions) and needs to operate in real time?
- Which algorithm or algorithms would be most appropriate for planning in very large domains (e.g., planning delivery routes for all UPS drivers in the U.S. on a given day)
- Which algorithm or algorithms would be most appropriate for planning problems where it is important to find only optimal plans?

Good work!

- About Greedy algorithm, indeed it's a good choice. Now, it's not guaranteed to be optimal. If we're forced to operate in real time with an optimal algorithm we may choose an uninformed search like BFS or UCS.
- Be careful on the second answer: A *with any planning graph heuristics and all uninformed searches do not scale well and should not be used. Remember that A* takes into account path cost so far and estimated distance to the goal. Greedy algorithm has the best scaling characteristics (it *only* takes into account the distance to the goal), and you may use a heuristic like planning graph or ignore preconditions. With a greedy algorithm, you sacrifice optimality per scalability.
- Yes, A* and BFS, UCS are guaranteed to be optimal.