**Artificial Intelligence Nanodegree Program**

**Project#2: Build a Forward-Planning Agent**

**March 12th, 2022**

**(1) Planning Graph Implementation**

* **my\_planning\_graph.py**

**(2) Heuristic Implementation**

* **my\_planning\_graph.py**

**(3) Experimental Results & Report**

**(a)** **Analyze the results as a function of domain size, search algorithm, and heuristic.**

<Definition of 11 Search Algorithms>

The number of Search Algorithm is defined as below:

1 [["breadth\_first\_search", breadth\_first\_search, ""],

2 ['depth\_first\_graph\_search', depth\_first\_graph\_search, ""],

3 ['uniform\_cost\_search', uniform\_cost\_search, ""],

4 ['greedy\_best\_first\_graph\_search',greedy\_best\_first\_graph\_search, 'h\_unmet\_goals'],

5  ['greedy\_best\_first\_graph\_search',greedy\_best\_first\_graph\_search, 'h\_pg\_levelsum'],

6  ['greedy\_best\_first\_graph\_search',greedy\_best\_first\_graph\_search, 'h\_pg\_maxlevel'],

7  ['greedy\_best\_first\_graph\_search',greedy\_best\_first\_graph\_search, 'h\_pg\_setlevel'],

8 ['astar\_search', astar\_search, 'h\_unmet\_goals'],

9  ['astar\_search', astar\_search, 'h\_pg\_levelsum'],

10 ['astar\_search', astar\_search, 'h\_pg\_maxlevel'],

11 ['astar\_search', astar\_search, 'h\_pg\_setlevel']

<Definition of 4 Problems>

Problem is defined in air\_cargo\_problems.py as below:



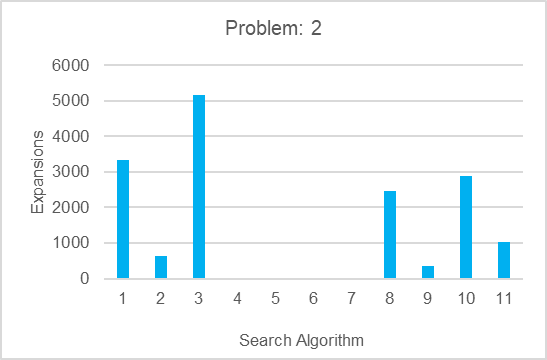
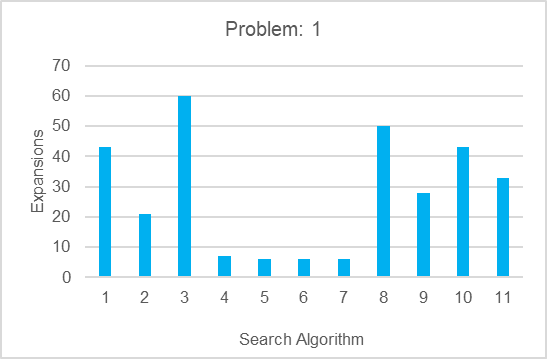
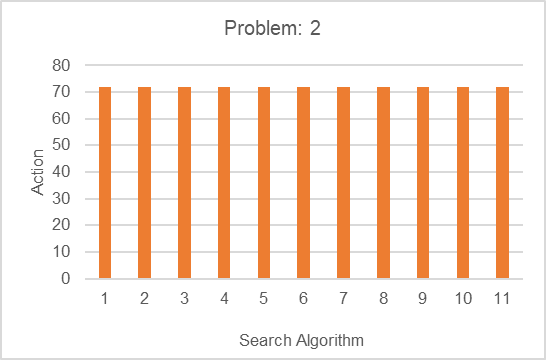
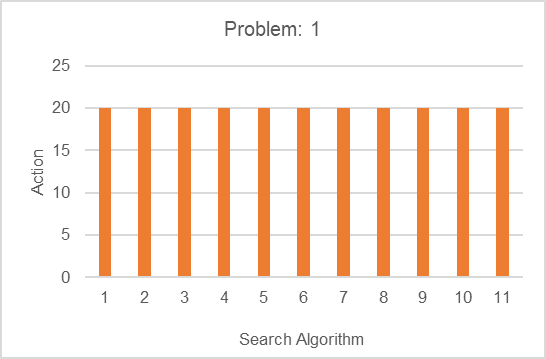
(I skip writing down every initial positions and goals.)

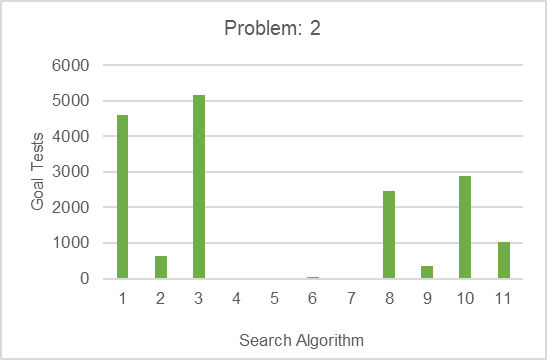
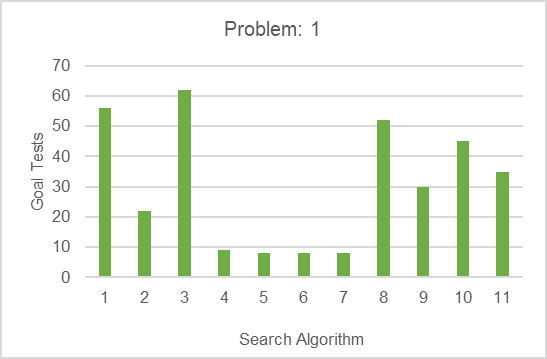
The search complexity, search time and the optimality as a function of domain size, search algorithm, and heuristic are shown below.

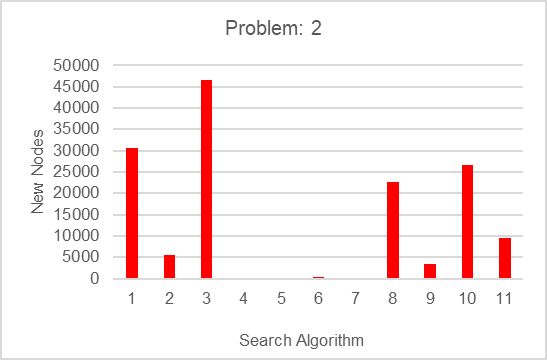
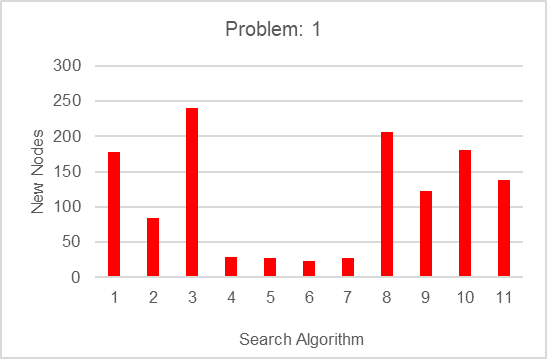
<Charts and Tables>

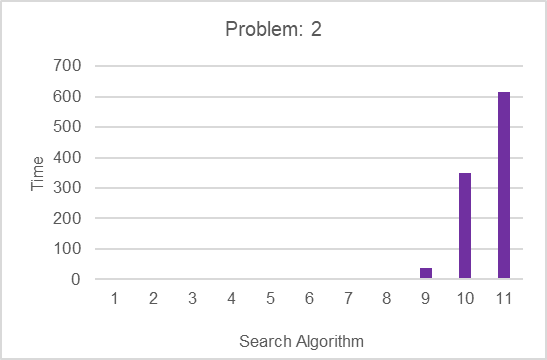
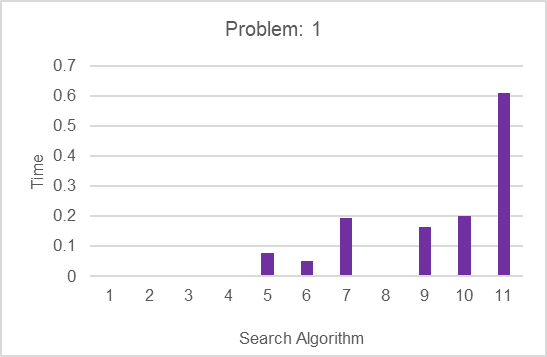
The chart and table includes data for all search & heuristic combinations for air cargo problems 1 and 2 are shown in the following table and graphs.

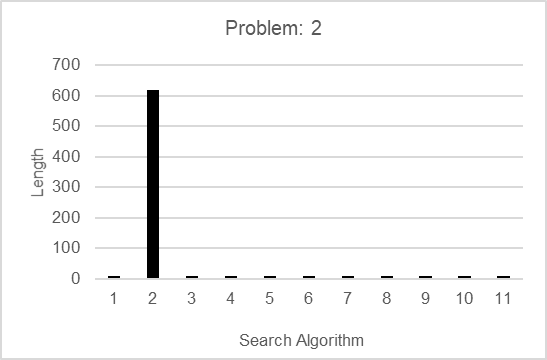
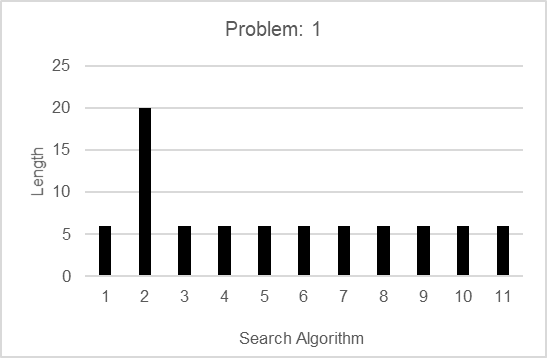






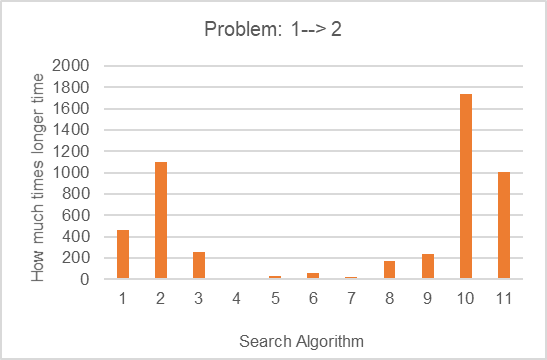






<1st Analysis>

* Only No.2 (depth first graph search) has big length (20) compared with other length (6) in problem1. In No.2, it became more than 30 times (619) even though other length became only 1.5 times (9) length. It might be memory shortage issue if the problem becomes more complicated.
* All of number of Actions became 3.6 times bigger from problem1: 20 to problem2: 72, but the Time became much longer in all of the search algorithm as below. At worst, in No.10 (A\* search, 'h\_pg\_maxlevel') Time became from problem1: 0.2sec to problem2: 348sec, which means 1736 times longer! At best, in No.4 (greedy best first graph search, 'h\_unmet\_goals') Time became from problem1: 0.0008sec to problem2: 0.007sec, which means 9 times longer.



<Algorithm Selection for problem3, 4>

I selected the following search algorithms:

* Equal or more than one uninformed search:

I should select one from No.1, 2, 3.

Action number and Time are not so different, but No.2 has the least Expansions, Goal Tests, New Nodes and Time, so **I chose No.2.**However, the length of No.2 is much bigger than other 2, which means that it would be memory shortage issue in the more complicated problem.

So for now, **I also chose No.3**.

* Two heuristics with greedy best first search:

I should select two from No.4, 5, 6, 7.

No.4 is much faster than others in problem 1 and 2, so **I chose No.4**.

There are not so difference between No. 5, 6, 7 in problem 1, but No.5 is faster than No.6, 7 in problem 2, so **I chose No.5**.

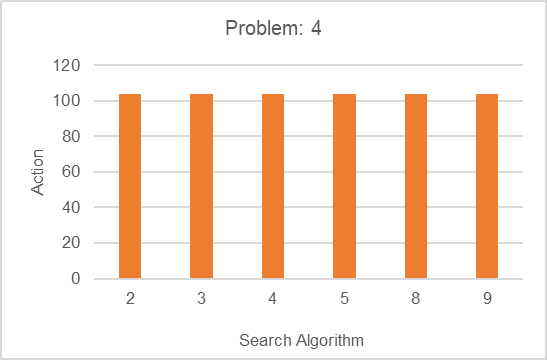
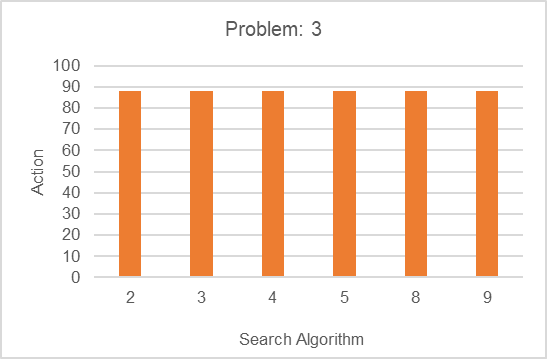
* Two heuristics with A\*:

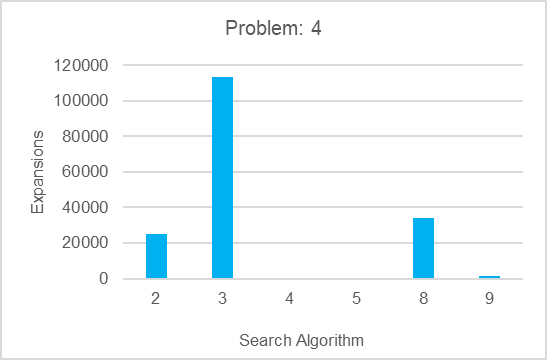
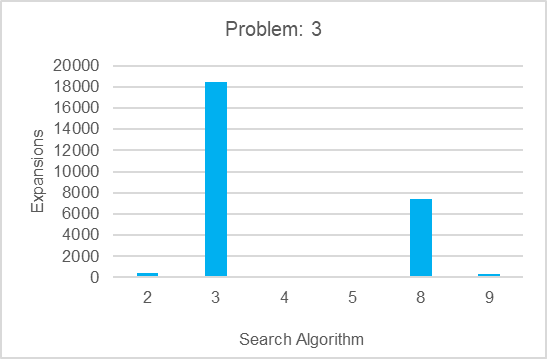
I should select two from No.8, 9, 10, 11.

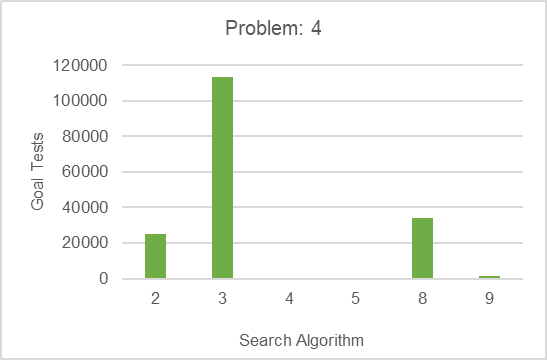
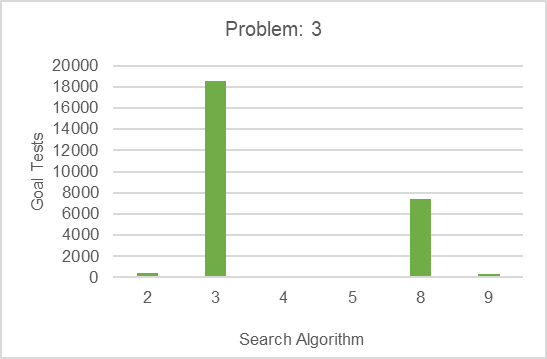
No.10, 11 are much slower than No.8, 9, so **I chose No.8, 9**.

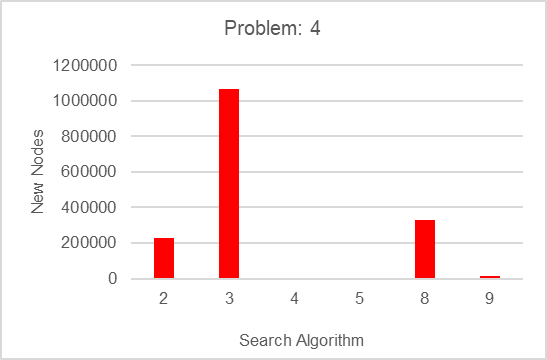
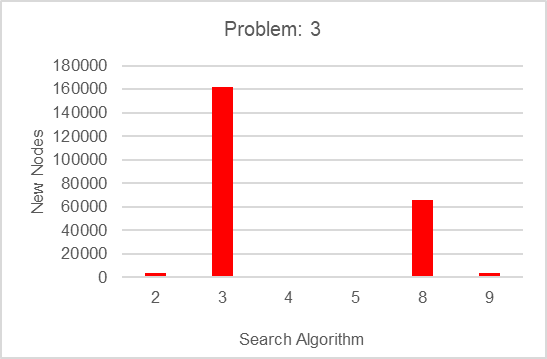
The chart and table includes data for selected search & heuristic combinations for air cargo problems 3 and 4 are shown in the following table and graphs.

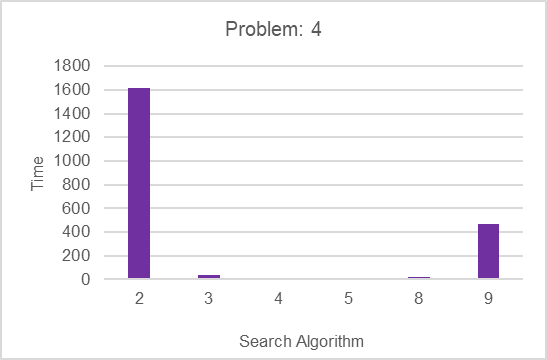
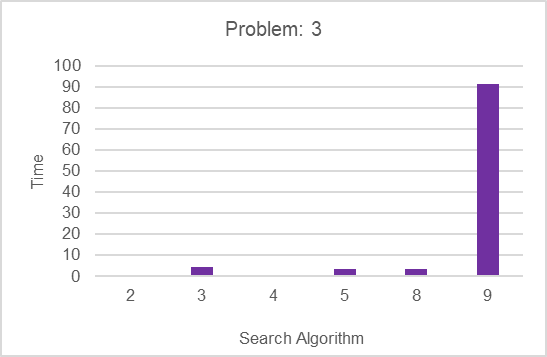


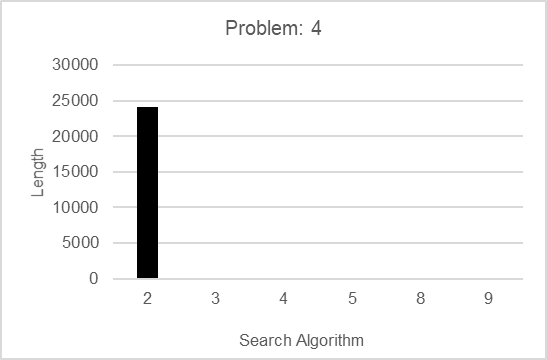
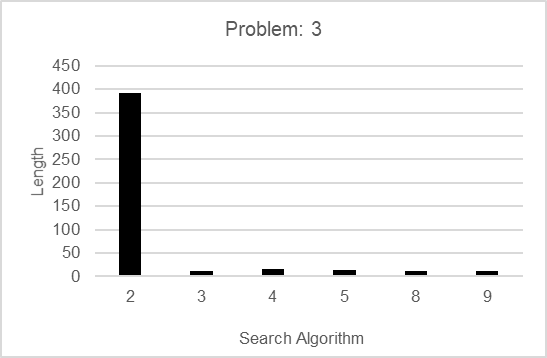












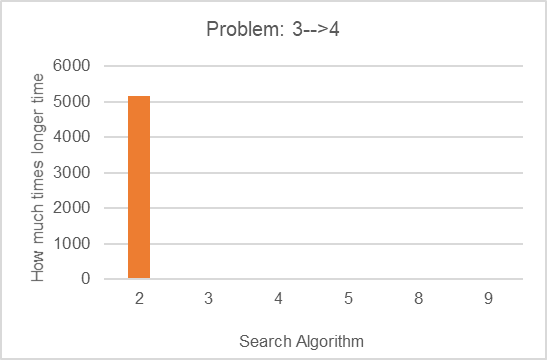
<2nd Analysis>

* No.2 (depth first graph search) is still fast (about 0.3sec) in proglem3, but it’s the slowest (about 1613sec) algorithm in these 6 algorithm.   
  The number of Actions became from 88 to 104 (about 1.2 times bigger), but the Length became from 392 to 24132 exponentially (about 62 times bigger).

That’s why it became so slow more than exponentially as below.

On the contrary, other algorithm became slower only 2 ~ 8 times from

Problem 3 to 4.



**(b) Report answers all required questions**

I answer the following 3 questions which are written in the rubric.

**- 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?**

🡪 I think it’s No.2 (depth first graph search) No.4 (greedy\_best\_first\_graph\_search, h\_unmet\_goals) because they are fast in a very restricted domain such as problem 1.

**- 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)**

🡪 I think it’s No.4 (greedy\_best\_first\_graph\_search, h\_unmet\_goals) because it’s the fastest algorithm in a very large domain such as problem 4.

**- Which algorithm or algorithms would be most appropriate for planning problems where it is important to find only optimal plans?**

🡪 I think it’s Greedy Best First Graph Search (No.4, 5) because it not only is fastest but also has smallest Expansions, Goal Tests, New Nodes compared with other algorithms.

It’s intuitively understandable as I learned in the Lesson as below.

