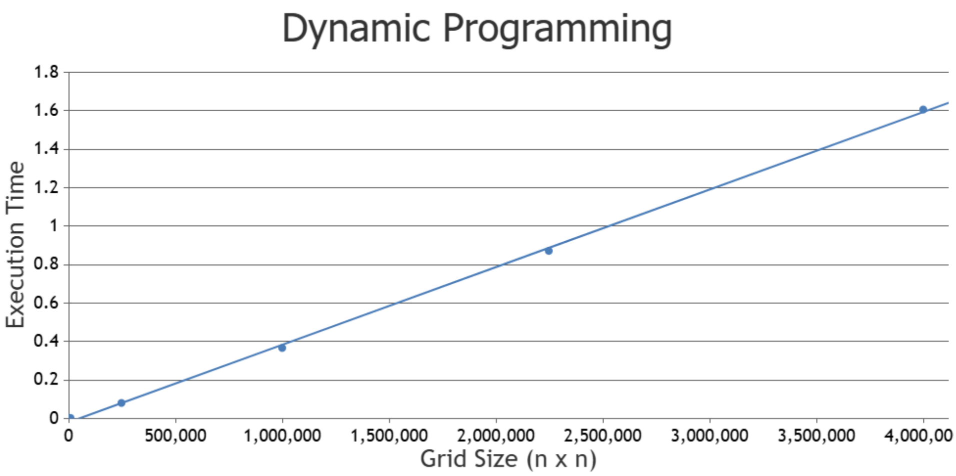
Project 2 Report - Opponent Avoidance Problem

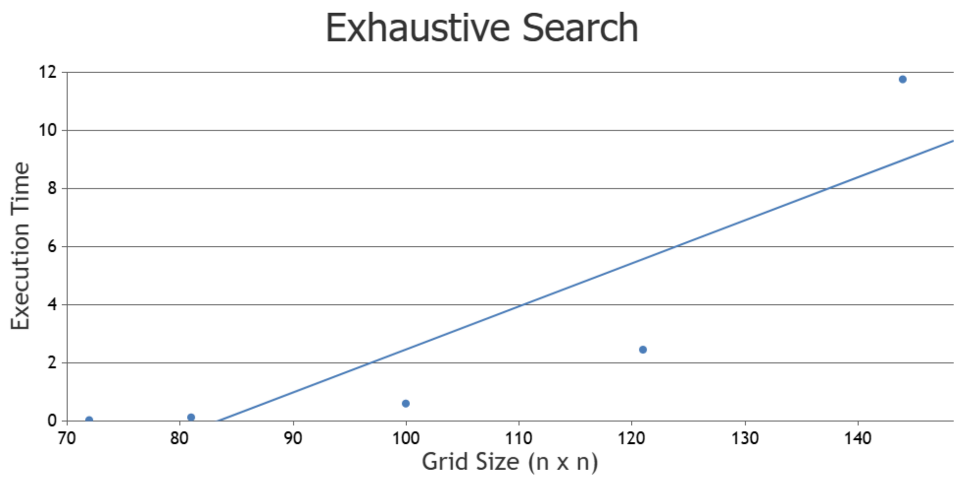
Chanran Kim (shining04@csu.fullerton.edu)
Brian Alvarez (briandalvarez@csu.fullerton.edu)
Christopher Contreras (CContreras71@csu.fullerton.edu)
Vinh Nguyen(vinhgod123@csu.fullerton.edu)

A. Scatter plots for each of two algorithms.

```
Timing DP algorithm...
Using dp algorithm, grid size 100 x 100 took 0.003000 seconds. Paths: 0
Using dp algorithm, grid size 500 x 500 took 0.080715 seconds. Paths: 0
Using dp algorithm, grid size 1000 x 1000 took 0.366848 seconds. Paths: 0
Using dp algorithm, grid size 1500 x 1500 took 0.871498 seconds. Paths: 0
Using dp algorithm, grid size 2000 x 2000 took 1.606287 seconds. Paths: 0

Timing Exhaustive algorithm...
Using exhaustive algorithm, grid size 8 x 8 took 0.030898 seconds. Paths: 0
Using exhaustive algorithm, grid size 9 x 9 took 0.125591 seconds. Paths: 0
Using exhaustive algorithm, grid size 10 x 10 took 0.601516 seconds. Paths: 0
Using exhaustive algorithm, grid size 11 x 11 took 2.455963 seconds. Paths: 0
Using exhaustive algorithm, grid size 12 x 12 took 11.759722 seconds. Paths: 0
```





B. Answers to the questions (Each answer should be at least one complete sentence.)

I. Is there a noticeable difference in the performance of the two algorithms?

There is a noticeable different in performance as Dynamic programming algorithm perform a lot better when the grid size increase. On the other hand, Exhaustive search peak 12 seconds when it was barely 140 grid size.

II. According to your experimental observation, which of the implementation is faster, and by how much?

The implementation that is faster and even more efficient would be the Dynanic programming. Dynamic programming is faster by huge amount. As you can see in the output and the scatter plots, even at 4000000 grid size it is still faster than Exhaustive search at 144 grid size by a whole 10 seconds.

III. Are your empirical analyses consistent with the predicted big-O efficiency class for each algorithm? Justify your answer.

For the exhaustive search, our analyse is consistent wit the predicted big-0 efficiency because as you can see, the ES has a exponential growth which demonstrate a high complexity time faulting in a big-0 notation of (2^n). It is too slow as the grid number move up. As for dynamic programing, it demosnrtae a polynomial growth which resulted in slower time as the grid number move up. Therefore, the notation of dynamic programming is (n^2).

IV. Is this evidence consistent or inconsistent with hypothesis 1? Justify your answer.

With the evidence that we've collected, we can support that the hypothesis is consistent. The reason for this is Exhaust search is a 0(2^n) notation and therefore implement an exponential running times. Which means that it would take its time to search for all possible solution resulting in a longer running time. It double checked every solution and eventually resulting in the most optimal solution.

V. Is this evidence consistent or inconsistent with hypothesis 2? Justify your answer

With all the evidence that we've collected, we believed that the hypothesis would be consistent with our evidence. Since, Exhaustive search take extremely long for any algorithm to use as it is a 0(2^n). Therefore, it would hinder any program that tried to use this algorithm for practical use.