Project 3

Sean Large G35125812 | sean.large1@gmail.com

1. **Problem Statement**

Magical Eggs, Cellphone drop test. Given a number of K eggs and N floors, return the amount of moves required to find the highest floor before the egg breaks.

1. **Implementation Characteristics**

Thinking in terms of m (moves), we can use dynamic programming to store results from moves to K (eggs) and reduce the amount of floors to choose from with each move. With each pass we can increase m and return m once we reach storage[K] = N. One way to implement this algorithm results in O(N^2 \* K) but this solution should be O(KlogN)

1. **Experimental Analysis**

## Programe Listing

See code in github or in blackboard [https://github.com/seanlarge/MagicEgg.git](https://github.com/seanlarge/Staircase.git)

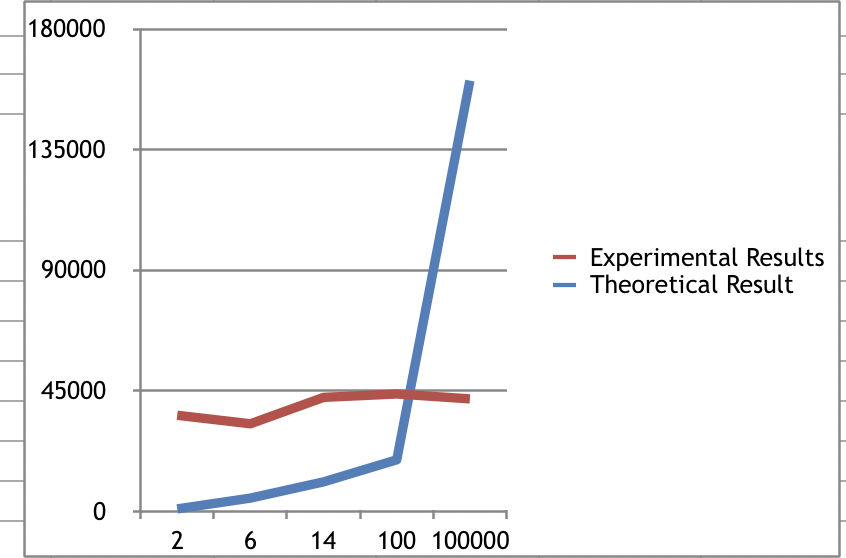
## Data Normalization Notes

###### Yes, 12.259143320692. I divided the average of experimental by the average of theoretical from table below

## Output Numerical Data

|  |  |  |
| --- | --- | --- |
| K,N | Experimental | Theoretical |
| 1, 2 | 35902 nanoseconds | 0.30102999566 |
| 2, 6 | 32727 nanoseconds | 1.55630250077 |
| 3,14 | 42576 nanoseconds | 3.43838410703 |
| 3,100 | 43847 nanoseconds | 6 |
| 10, 100000 | 42011 nanoseconds | 50 |

## Graph



Graph

## Graph Observations

Something looks off with the way my code (and the graph) runs but the results are correct. My runtime is inconsistent. Sometimes larger numbers will run faster than smaller.

1. **Conclusions**

The runtime is O(KlogN)