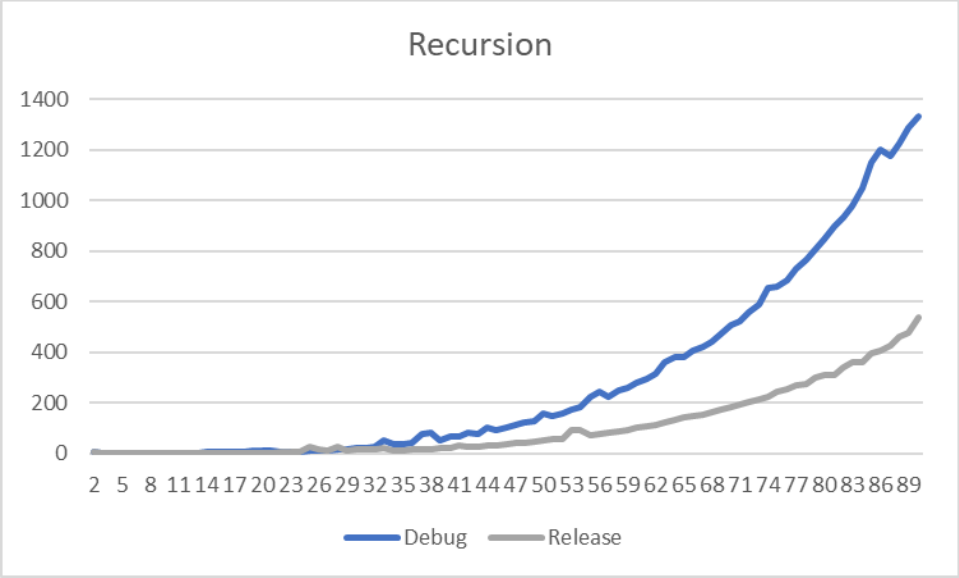
Performance Report for Assignment 5 by Dan Moraru – 2138261

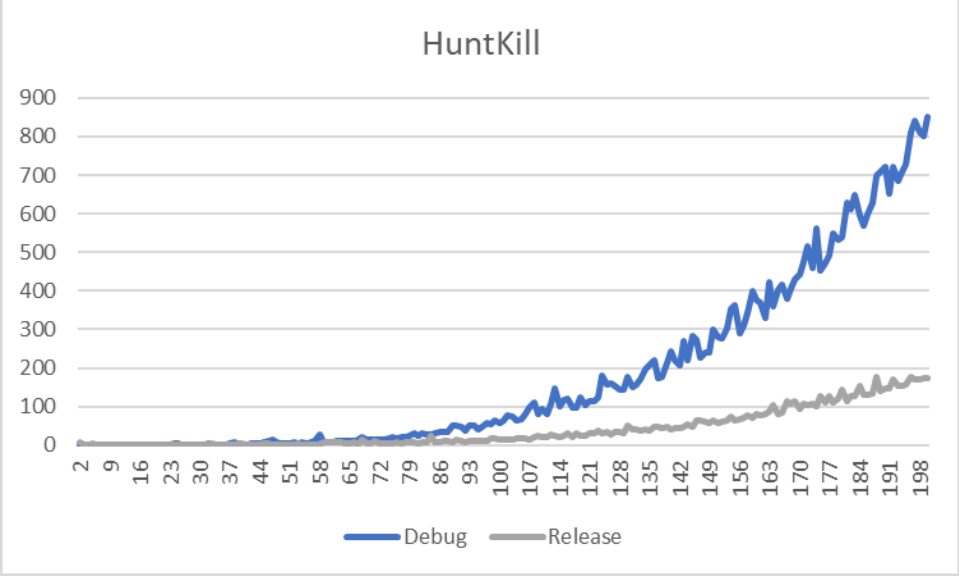
Used computer labs, specs:

- Intel(R) Core(TM) i9-10900X CPU @ 3.70GHz 3.70 GHz
- 64.0 GB of RAM
- Windows 10 Enterprise, 19045.3324

Results, x = square size of maze, y = time in ms :



Clearly it is a O(n^2) notation. The limit for me was around the size of 90x90.



Clearly it is a O(n^2) notation, it fluctuates a bit, but the notation is still visible. The limit for me was around the size of 200x200.

Comparing Release and Debug:

- At first, the HuntKill algorithm is much faster than Recursion, on both Release and Debug
- Release improves speed a lot, for recursion the cap is [Debug: 1300, Release: 530] and for huntkill [Debug:850, Release:180]
- Sizes are also different, Recursion: 90 & HuntKill: 200, before crashing

Analysis:

Upon looking at my recursion algorithm, it is already pretty optimized. The only changes I could think of is to save my list of direction enums as a field, so that I would not have to set it everytime I call the method, and implement the parallel for loop instead of the regular for loop, since the recursion does not really need to keep track of the previous iterations.

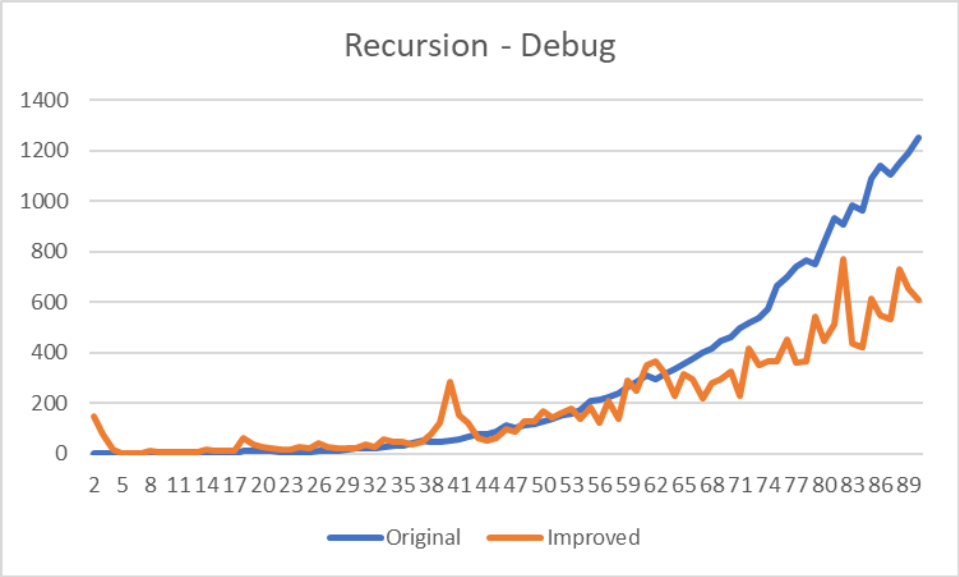
However, when looking at my HuntKill algorithm, I realized that I could do much more improvements such as creating an availableVectors list instead of looping through everything while hunting, and setting it as a HashSet instead of a list.

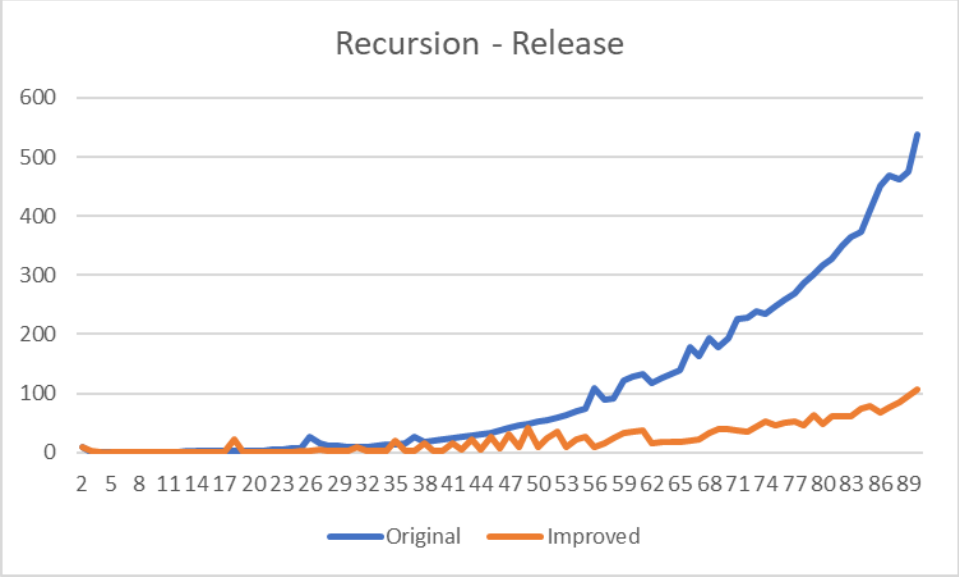
After-Analysis:

I decided to improve both, because I was curious about the results. In the end I realized that my improvements to my recursion algorithm were very good, almost reducing the time in half.

However, my improvements on my HuntKill algorithm were actually slower, which is weird since I thought it would make it faster. My thoughts are that in order to setup my implementations, like creating the availableVectors list from looping through the height and width, actually use more resources and time than if I were to do nothing about it, which hinders the speed.

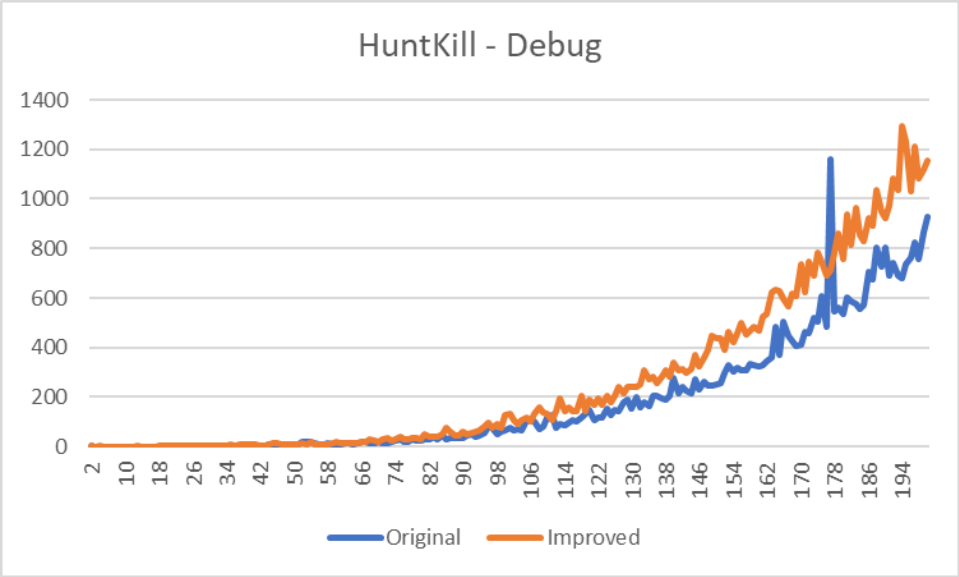
Improved Recursion Algorithm, x = square size of maze, y = time in ms:

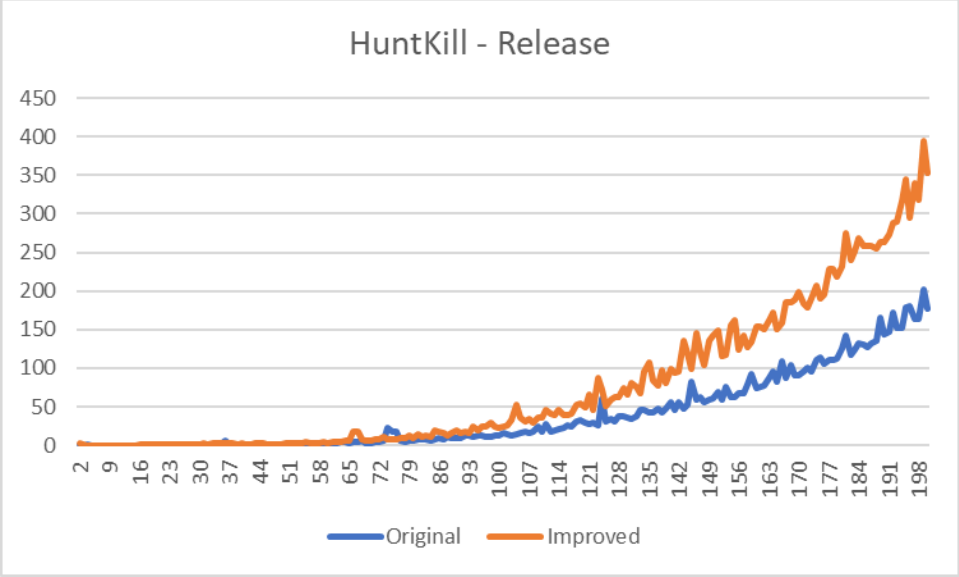




The fluctuations that happen in the improved version is due to the parallelized for loop, despite that, it is clear to be a $O(n^2)$

Improved HuntKill Algorithm, x = square size of maze, y = time in ms:





The fluctuations don't make it hard to tell that this is indeed a $O(n^2)$