CSCE 221 Report 1

Carter Cummings

UIN: 630003063

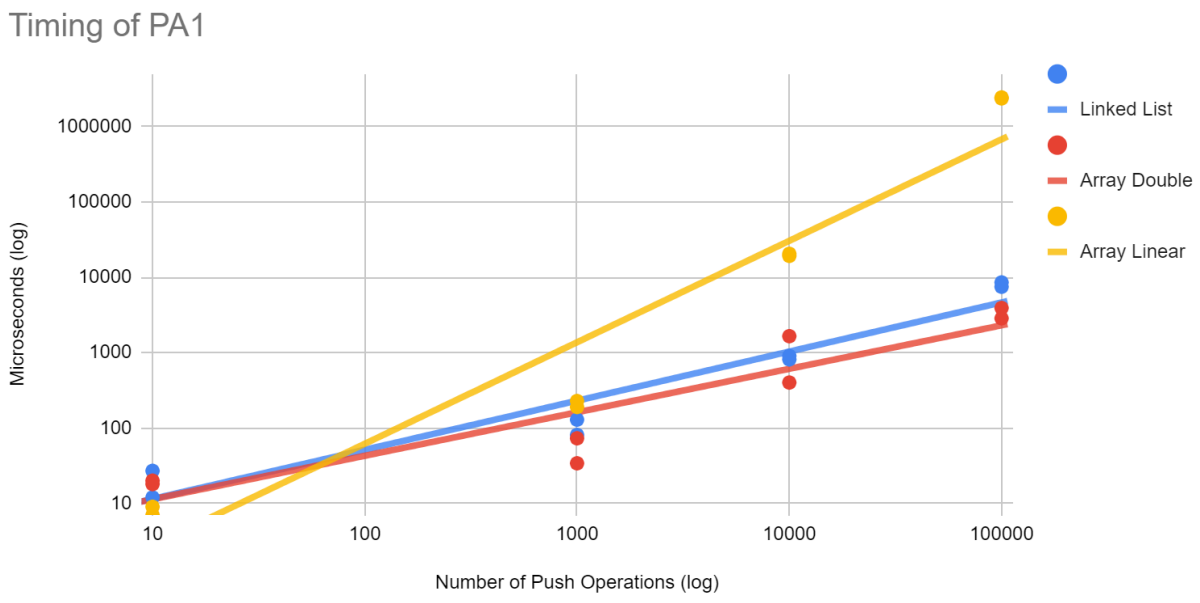
Introduction:

For this assignment it is important to understand how to create programs that avoid using linear or worse time. This example was given by taking three different approaches at the same problem to understand what is effective at reducing the run time of a program. Thus this assignment was about creating the linked list program as well as the linear increasing and the doubling stack then comparing how they run.

Analysis:

For this assignment the largest use of time and difference between the three approaches is the “push” to stack function. This is due to having to allocate more memory for the stack if the capacity has been reached. The first and slowest approach is the array that grows linearly; this will always take linear time dramatically slowing down when even getting to lengths above ten thousand taking close to second, further detail can be found in the results section. Then after this which are much more comparable are doubling array and linked list. As each node of the linked list takes the same amount of time to add it is still linear however this linear time is much faster and the time growth is much much slower than the array linear. The fastest at the high end is array doubling. This will end up taking logarithmic time which as the orders of magnitude of the data grow would become clearly the fastest.

Results:



Two trials were done for each stack type and the time was collected to do the number of push operations in microseconds. The first run was thrown out to allow the optimizer to run the code

better. The graph above is also in logarithmic x and y scales this just allowed for better visualization of the growth of the data. The linked list and array double grow at similar rates and with more test runs at more points this data could have a much higher resolution however as of now this graph is able to prove the point of why optimization is important. At 100k operations linear already takes upwards of 2 or 3 seconds and this would only get worse. While array double is still only taking about 5000 microseconds. This shows how much better it is to have at worst log(O) time so that operations can grow a lot faster than time taken.