PA1: Stack

Ilan Sela

Texas A&M University
College Station, TX 77843, US.

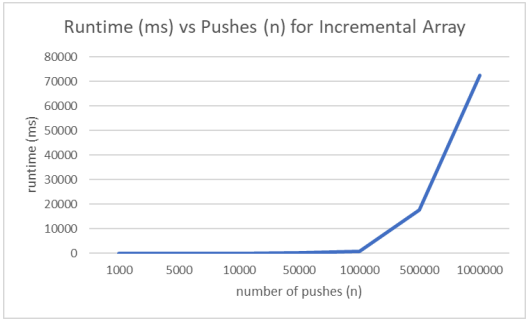
1. Introduction

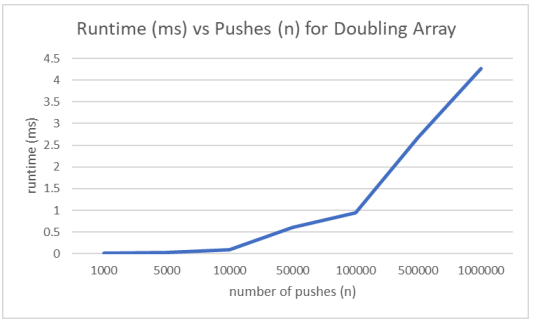
The purpose of this assignment is to familiarize students with the stack ADT. This was done by implementing the ADT with array and linked list data structures. The stack ADT is first in last out (FILO) and involves size, empty, top, push, and pop functions.

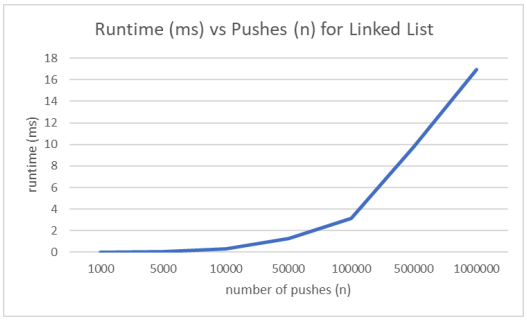
2. Theoretical Analysis

The push operation in the stack ADT adds an element to the front of the stack. The three strategies used in this lab are: incrementing array, doubling array, and linked list. The incrementing array has average O(n) complexity. This is because its worst case is O(n^2), which happens when you grow the array but only need it for one item, and then amortized, the average becomes O(n^2)/n = O(n). The doubling array has average O(1) complexity. Its worst case is O(n) and its average becomes O(n)/n = O(1). This makes sense as the incrementing array will need to copy the old array to the new array when resizing more often than doubling (n/c versus log2(n) number of resizing). On par with the amortized doubling array is the linked list, as the push function's complexity is always O(1). This is because there is no need to loop or copy the list, we only need to replace the head; which is an advantage of linked lists. A disadvantage is that we need to store a pointer to the next element with the value for each allocation. An advantage of arrays is that each element is stored consecutively in memory, so it is quicker to access, whereas in linked lists the computer has to find the memory address location to retrieve what's stored. An advantage of an incremental array is that it will never waste memory, however, its runtime is very bad.

3. Results and Analysis







The doubling array stack performs the best while the incremental array performs the worst, as predicted before. The number of pushes does affect the difference in runtime between each implementation. For a low number of pushes (<5000), the runtime has little to no difference. Due to the different growth rates, as the number of pushes increases, the cumulative effects give a more pronounced difference. The experimental and theoretical results match for the most part. Big-oh for the double array and linked list is the same, however, that is not the case in the experimental results for reasons mentioned in the theoretical analysis (such as memory allocation differences).