PROG 20799

Assignment 3

Jason Beattie

Manuel Sanchez

Part 1

# Stack implementation using Single Link List.

Part 2

# Performance comparison between Bubble sorting, Selection sorting and Quick sorting.

**Time records taken by type of execution (in seconds).**

|  |  |  |
| --- | --- | --- |
|  | ***Sorted Data Set Timing*** | ***Random Dataset Timing*** |
| ***Selection Sort*** | 0.234000 | 0.203000 |
| ***Bubble Sort*** | 0.828000 | 0.672000 |
| ***Quick sort*** | 0.016000 | 0.547000 |

Note: measurements taken right after clean and build the program to get results for Quick sorting algorithm.

**Why there are some differences between theoretical and practical results.**

Considering two sources of unsorted data to compare these three sorting algorithm this are our thoughts.

Bubble sort iterates proportionally to the input size so the results for both source of data are close in time, not showing considerable differences. The worst case of this algorithm is a time O(n^2).

Selection looks pretty much the same as Bubble sort. Looking the results for both source of data they are not show big differences. The worst time for this algorithm is O(n^2).

Conversely, Quick Sort algorithm shows noticeable different results depending on the source considered. This algorithm presents a worst case when the series has the most unbalanced partition with O(n^2) and the best when it is evenly balanced (n log n). In the latter, the time is shorter and that seems to be the case for our results. While the source file contains unsorted numbers but all of them has 5 digits, the random array has different number of digits. So, it seems