Bonus Homework #9

David Tooley

Part 1:

Sort algorithm run on an array of the form [n,n-1,n-2,…..,1]

|  |  |  |  |
| --- | --- | --- | --- |
| n | Insertion Sort Actions | Merge Sort Actions | Quick Sort Actions |
| 100 | 15346 | 2374 | 1869 |
| 200 | 60696 | 5254 | 4125 |
| 400 | 241396 | 11541 | 9637 |
| 800 | 962796 | 25034 | 22153 |
| 1600 | 3845596 | 54074 | 50669 |
| 3200 | 15371196 | 116154 | 122421 |
| 6400 | 61462396 | 248314 | 226869 |

Insertion sort is an O(n2) algorithm, in this case each time n is doubled the number of actions taken is 4 times the previously needed amount, which is the expected correlation between these two events.

Merge sort and quick sort are both O(n log(n) ) functions. As expected for every time n is doubled the number of actions required grows by a slightly greater amount than two.

Part 2:

Sort algorithm run on an array of the form [n-4,n-3,n,n-1,n-2]

|  |  |  |  |
| --- | --- | --- | --- |
| n | Insertion Sort Actions | Merge Sort Actions | Quick Sort Actions |
| 100 | 4121 | 2068 | 1805 |
| 200 | 15746 | 4542 | 4428 |
| 400 | 61496 | 9890 | 9120 |
| 800 | 242996 | 21386 | 22880 |
| 1600 | 965996 | 45978 | 42393 |
| 3200 | 3851996 | 98362 | 109626 |
| 6400 | 15383996 | 209530 | 219804 |

In this case where the vector is not in reverse order it is shown that the number of actions for Insertion Sort and Merge Sort are smaller for every evaluation of n, while Quick Sort remains roughly the same. This is because Quick sort is while Insertion sort and Merge sort, can accomplish their tasks much quicker if they items are already partially sorted.

Source code: main.cpp

I am using Visual Studio 2012 version 6.1.7601 Service Pack 1 Build 7601

My operating system is Windows 7 Ultimate

Implementation of code:

First the code asks if you wish to run the sort algorithms with vector 1 or vector 2. Vector 1 is the vector being tested in part 1 that has the form [n,n-1,n-2,…..,1]. Vector 2 is the vector being tested in part 2 with the form [n-4,n-3,n,n-1,n-2]. Acceptable input is either 1 or 2. Next the code requests that you give it the number of elements to insert into the vector. Then the program will create the sorted version of the vector using Insertion Sort, Merge Sort, and Quick Sort and print the results out to the user.