

CS 302

Homework, Asst. #03

Purpose: Learn concepts regarding sort algorithms and sorting algorithm analysis.

Due: Tuesday (9/16) → Must be submitted on-line before class.

Points: Part A → 40 pts, Part B → 60 pts

Assignment – Part A:

Write a C++ program to implement the following sorting algorithms:

- Selection Sort¹
 - Use the standard selection sort algorithm as outlined on the Wikipedia page. The parameters include the array and the starting and ending index (instead of just the length), this allowing it to sort a subset of an array.
- Quick Sort²
 - Use the algorithm as outlined in the text, with the median of three selection for the pivot value. The algorithm calls an insertion sort which should be changed to the selection sort.
- Bubble Sort³
 - Use the optimized bubble sort algorithm as outlined on the Wikipedia page.



The Sorting Hat

Part B:

When completed, use the provided script to execute the program on a series of different counts of numbers (100,000, 200,000, ..., and, 1,000,000). The script will write the execution times to a text file. Enter the counts and times into a spreadsheet and create a line chart plot of the execution times for each algorithm. Refer to the example for how the plot should look.

Create and submit a write-up (open document or word format) with a write-up not too exceed 200 words including the following:

- Name, Assignment, Section
- Description of the machine used for obtaining the execution times.
- Copy of the chart.
- Explanation of the results, comparing the algorithms.
- Comments regarding the use of recursion (good, bad, n/a).

Note, execution times for each submittal will be different (possibly very different).

Once the program is working, create a copy and change the *generateRandomValues()* function to assign “**i+10**” (instead of *rand()*), thus creating a sorted list. Also, in the *quickSort()* function change the pivot assignment to **pivot = arr[right-1]** instead of using the *median3()* function. Execute the program with both the *bubbleSort* (-bs) and *quickSort* (-sq) functions with an -n value of 250,000. Include the results of this test and an explanation for results in the write-up.

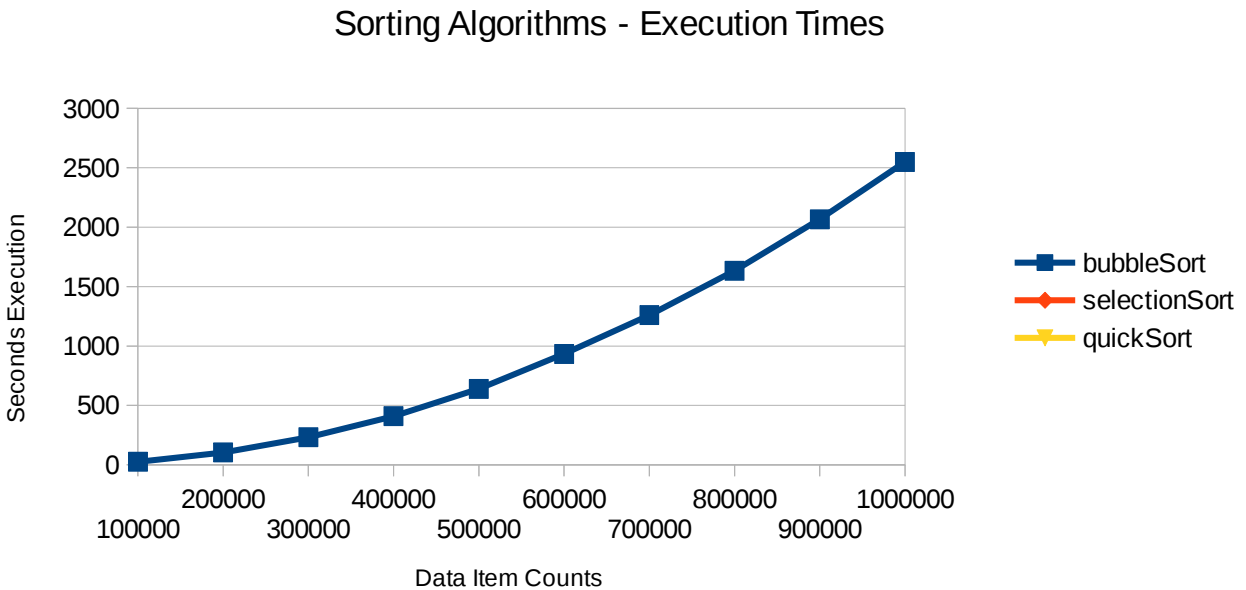
1 For more information, refer to: http://en.wikipedia.org/wiki/Selection_sort

2 For more information, refer to: <http://en.wikipedia.org/wiki/Quicksort>

3 For more information, refer to: http://en.wikipedia.org/wiki/Bubble_sort

Example Plot:

Below is an example of the execution times plot (excluding the quick select times). This incomplete example show to appropriate format.



The final chart should be complete and show the times for all three algorithms (instead of just one as shown in the example above).

For reference, the following link has a number of animation to help understand how each sort functions.
<https://www.cs.usfca.edu/~galles/visualization/ComparisonSort.html>

Submission:

When complete, submit:

- Part A → A copy of the **source file** via the class web page (assignment submission link) by class time on the due date.
- Part B → A copy of the write-up including the chart. Must use “.odt”, “.doc”, “.txt”, “.rtf”, “.docx”, or “.pdf” format.

Assignments received after the due date/time will not be accepted.

You may re-submit as many times as desired. Each new submission will require you to remove (delete) the previous submission.

Make sure your program includes the appropriate documentation. See Program Evaluation Criteria for CS 302 for additional information.