

Algorithms

Lab 07 (8%)

Topics: Sorting and Searching
(MCCME: <http://informatics.mccme.ru/>)

In each of problems of this lab besides writing corresponding program you have to analyze and characterize algorithm you use (Cormen “Algorithms Unlocked Chapters 2, 3)

Problem 01 (1%)

(Insertion Sort)

Write Java program to solve problem MCCME 232. Your program must have method `insertSort(int[] a)`. This method must be your implementation of insertion sort algorithm.

Problem 02 (2%)

(Merge Sort)

Write Java program to solve problem MCCME 766. Program must have method `mergeSort(int[] a)`. This method must be your implementation of merge sort algorithm.

Problem 03 (2%)

(Quick Sort)

Write Java program to solve problem MCCME 766. Program must have method `quickSort(int[] a)`. This method must be your implementation of quick sort algorithm.

Problem 04 (1%)

(Empirical comparison of performance of InsertSort, MergeSort and QuickSort sorting algorithms)

Write Java program to compare performance of methods `insertSort`, `mergeSort` and `quickSort` for sizes: 1000, 2000, 4000, 8000, ..., 64000

For each method your program must output the following table:

<Size of array>	<Milliseconds>	< Ratio (current time / previous time)>
-----------------	----------------	-----------------------------------------

Check your sorting methods for different (random, non-increasing, non-decreasing) orders of elements.

Give informal analysis of each algorithm and explain empirical results.

Problem 05 (1%)

(Binary Search: iterative version)

Write Java program to solve problem MCCME 2. Program must have iterative method `int binarySearch(int[] a, int key)`. This method must be similar to the corresponding method from Java library.

Problem 06 (1%)

(Binary Search: recursive version)

Write Java program to solve problem MCCME 2. Program must have recursive method `int binarySearch(int[] a, int key)`. This method must be similar to the corresponding method from Java library.