

Implementation of data structures and algorithms  
Short Project 3: Divide and Conquer

Version 1.0: Initial description (Wed, Sep 11).

**Due: 11:59 PM, Mon, Sep 23 2019.**

Submission procedure: same as usual.

**Team task:**

1. Implement and compare the running times of the following algorithms on randomly generated arrays: (a) Merge sort (take 1), (b) Merge sort (take 3), (c) Merge sort (take 4), (d) Merge sort (take 5). Do not run more than one algorithm in each trial. In each trial, run only one algorithm, for one value of  $n$ , 100 times in a loop, and taking the average time. Try the following values of  $n$ : 8M, 16M, 32M, 64M, ..., until you get out of memory exception. Submit a report with your observations.

**Practice task (optional):**

2. Use BigInteger class and write programs to compute  $f(n)$ , the  $n$ th Fibonacci number. Implement the  $O(n)$  algorithm and the  $O(\log n)$  algorithm. Compare their running times for the following values of  $n$ : 1000, 10000, 100000. Since printing the output takes a lot of time, stop the timer before starting to print the output. In each trial, run only one algorithm, for just one value of  $n$ .

3. Given an array of unique integers (no duplicates) and target  $x$ . Write a binary search that returns the index of the target if it is found. Otherwise, return the index where the target can be inserted without violating the sorted order of the array. Write the invariants before coding.