

# Assignment 1: CSCI 4310/6323

Instructor: Dr. Bin Fu. Due Feb. 11, 2021 (Thursday). Please type solution in Microsoft Word Format.

**Problem 1.** Give asymptotic upper and lower bounds for  $T(n)$  in each of the following recurrences. Assume that each  $T(n)$  is a constant for  $n \leq 2$ . Make your bounds as tight as possible, and justify your answers. a)  $T(n) = 6T(n/6) + n$ , b)  $T(n) = 4T(n/2) + n^3$ , c)  $T(n) = T(n-1) + n^2$ , d)  $T(n) = T(n-1) + \frac{1}{n}$ .

**Problem 2.** How many lines does the following program print? Write a recurrence and solve it. You may assume that  $n$  is a power of 2.

```
function f(n)
if (n > 1)
    print.line ("still going");
    f(n/2);
    f(n/2);
    f(n/2);
```

**Problem 3.** Let  $A[0 \dots n-1]$  be an array of  $n$  distinct integers. A pair  $(A[i], A[j])$  is said to be an inversion if these numbers are out of order, i.e.,  $i < j$  but  $A[i] > A[j]$ . Design a  $O(n \log n)$  time algorithm for counting the number of inversions.

**Problem 4.** Design a C++-program to implement the following functions: a) the function `int bubblesort(int *a, int size)` for bubble sorting, b) the function `int mergesort(int *a, int size)` for merge sorting, c) the function `int quicksort(int *a, int size)` for quick sorting, which is described in the textbook, d) the function for generating array of random elements: `int generate(int *a, int size)` which calls the function `rand()` in C++, and e) Test both bubble sorting and merge sorting with 10, 100, 1000, 10000, 100000, 1000,000, and 4000,000 integers. The integers are from the array generated by part d). Calculate the time spent in calling each sorting. You may use a function in `<time.h>` to get the current time. Draw curves to compare the speed performance among the three sorting functions. The merge sorting and quick sorting algorithms implementation must be described by recursions. You are expected to define a global array for holding elements in both merging and quick sorting. Otherwise, it may cause a lot extra memory in recursion. Hint 1: use the following format to calculate the time cost for bubble sort.

```
{
time1=Get the current time (call a function in <time.h>);
bubblesort(.);
me2=Get the current time (call a function <time.h>);
timeCost = the difference between time1 and tim2;
}
```

Print your program and test results.