CS 141, Fall 2019
Assignment 2
Posted: October 14th, 2019 Due: October 28, 2019, 11:59pm

Notice

• Include your full name and student ID in your solution
• You are expected to work on this assignment on your own
• Use pseudocode, Python-like or English to describe your algorithms. Absolutely no C++/C/Java
• When designing an algorithm, you are allowed to use any algorithm or data structure we explained in class, without giving its details, unless the question specifically requires that you give such details
• Always remember to analyze the time complexity of your algorithms
• Homework has to be submitted electronically on iLearn by the deadline. Late submission allowed for 20% penalty for a calendar day.

Problem 1. (15 points) A is an Array of integers, so A=a1, a2, a3, ... an. Design an algorithm to find a peak element in A. An array element is peak if it is NOT smaller than its neighbors. For corner elements, we need to consider only one neighbor. For example, for A=5, 10, 20, 15, 20 is the only peak element. For A=10, 20, 30, 40, 50, 50 is the only peak element. For A=100, 80, 60, 50, 20, 100 is the only peak element. For A=10, 20, 15, 2, 23, 90, 67, there are two peak elements: 20 and 90. A=10, 10, 10,10, all elements are peak elements. Note that we need to find any one peak element. Report the complexity of your algorithm. Now, design a divide and conquer algorithm that runs in $O(\log n)$ to find a peak element in A.

Problem 2. (15 points) Analyze the recursive version of the fibonacci series. Define the problem, write the algorithm and give the complexity analysis

Problem 3. (25 points)
1. $T(n) = 4T(n/2) + \frac{n^2}{\log n}$
2. $T(n) = T(n/3) + T(2n/3) + n$
3. $T(n) = 4T(n/3) + n \log_4 4$

Problem 4. (15 points) Given an array of numbers $X_1 = \{x_1, x_2, ..., x_n\}$ an exchanged pair in X is a pair $x_i, x_j$ such that $i < j$ and $x_i > x_j$. Note that an element $x_i$ can be part of up to $n-1$ exchanged pairs, and that the maximal possible number of exchanged
pairs in X is \( n(n-1)/2 \), which is achieved if the array is sorted in descending order. Give a divide-and-conquer algorithm that counts the number of exchanged pairs in X in \( O(n\log n) \) time.

**Problem 5.** (15 points) You are given two sorted arrays of integers A and B of size m and n respectively. Describe a divide and conquer algorithm that takes \( O(\log k) \) time for computing the \( k-th \) smallest element in the union of the two arrays. Assume that the integers in both arrays are distinct, and no integers in A appear in B (and vice versa). Explain carefully why your algorithm takes \( O(\log k) \) time. Hint: The \( k-th \) smallest element in the union of the two arrays has to be contained in A[1...k] and B[1...k].

**Problem 6.** (15 points) Imagine you are using a sequential memory access (SAM) machine, write an algorithm to reverse a given array of integers A. For example, if A = (3, 2, 1, 4, 5), the algorithm should output Reverse(A) = (5, 4, 1, 2, 3). Assume A is stored at the very beginning of SAM memory. Analyze the time complexity of this algorithm using SAM machine and Random Access Memory (RAM) machine.

Note: Sequential access memory (SAM) machine is a computing machine that reads stored data slots in a sequence. So, to read A(4), the machine must go over slots of A(1), A(2), and A(3). This is in contrast to random access memory (RAM) where data can be accessed in any order. For example, to access A(4), RAM machines can reach A(4) directly without traversing A(1) through A(4).