CSC-421 Applied Algorithms and Structures
Spring 2021-22

Instructor: Iyad Kanj
Office: CDM 832
Phone: (312) 362-5558
Email: ikanj@cs.depaul.edu
Office Hours (Office/Zoom): Monday 4:40-5:40 & Wednesday 1:00-3:00
Course Website: https://d2l.depaul.edu/

Sample Midterm Exam

I. (20 points) Give a tight solution to the following recurrence:
$T(n) = T(n-1) + n/2$, and $T(1) = 1$.

II. (20 points) Let $A[1..n]$ and $B[1..n]$ be two sorted arrays of numbers. We can easily output the k-th smallest element in A in constant time by just outputting $A[k]$. Similarly, we can find the k-th smallest element in B. Give a $O(\lg k)$ time divide-and-conquer algorithm to find the k-th smallest element overall – i.e., the k-th smallest in the union of A and B. Explain why your algorithm works. You can assume all elements in A and B are different.

III. (20 points) Given an array A of n numbers, we would like to find the minimum and the maximum numbers in A. Assume that n is a power of 2, that is, $n = 2^k$ for some k.

1. Design a divide-and-conquer algorithm for the problem that makes at most $3n/2$ comparisons.
2. Design an iterative algorithm for the problem that makes at most $3n/2$ comparisons.

IV. (20 points) Give an $O(n \lg m)$ time algorithm that computes the union $A \cup B$ of an m-element set A and an n-element set B, where $m \leq n$. The sets A and B contain numbers and are represented as arrays. The arrays are not necessarily sorted and there are no duplicates in either array. The output should be an array that represents the union $A \cup B$ and that contains no duplicates. (A and B may contain common elements.)

V. (20 points) Suppose you have m sorted arrays, each with n elements, and you want to combine them into a single sorted array of $m \cdot n$ elements. Given an $O(nm \lg m)$-time algorithm to solve this problem. Explain the steps of your algorithm and analyze its running time.