CSC-421 Applied Algorithms and Structures
Winter 2019

Instructor: Iyad Kanj
Office: CDM 832
Phone: (312) 362-5558
Email: `ikanj@cs.depaul.edu`
Office Hours: Monday & Wednesday 4:00-5:30
Course Website: https://d2l.depaul.edu/

Sample Midterm Exam

I. Give an $O(n \log n)$-time algorithm that takes an array A of n numbers that may contain duplicates (elements that are equal), and removes all duplicates from A. That is, each number in A should appear **exactly once** in the resulting array, after the duplicates have been removed.

II. 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 nonnegative integer k.

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

III. For each of the five questions below, circle the correct answer.

(i) The most efficient algorithm for sorting an arbitrary/general array of n numbers has a worst-case running time of:

(A) $\Theta(\lg n)$
(B) $\Theta(n)$
(C) $\Theta(n \lg n)$
(D) $\Theta(n^2)$
(E) None of the above

(ii) The most efficient algorithm for searching (for a given number) a sorted array of n (arbitrary) numbers has a worst-case running time of:

(A) $\Theta(\lg n)$

(B) $\Theta(n)$

(C) $\Theta(n \lg n)$

(D) It depends on whether the array is sorted in increasing or decreasing order

(E) None of the above

(iii) The most efficient algorithm for searching (for a given number) an arbitray/general array of n numbers has a worst-case running time of:

(A) $\Theta(\lg n)$

(B) $\Theta(n)$

(C) $\Theta(n \lg n)$

(D) It depends on whether the array is sorted in increasing or decreasing order

(E) None of the above

(iv) If the Travelling Salesman Problem (TSP) is solvable in polynomial time then P=NP.

(A) True

(B) False

(C) One cannot make a conclusion based on the above statement

(v) Each of Independent Set and Travelling Salesman Problem is polynomial-time reducible to the other. (That is, Independent Set $\leq_P$ Travelling Salesman Problem and Travelling Salesman Problem $\leq_P$ Independent Set.)

(A) True

(B) Only Independent Set is polynomial-time reducible to Travelling Salesman Problem

(C) Only Travelling Salesman Problem is polynomial-time reducible to Independent Set

(D) None of the two problems is polynomial-time reducible to the other

IV. Let $A[1..n]$ and $B[1..n]$ be two sorted arrays. 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; that is, the k-th smallest in the union of A and B.

V. Let X be a finite set and $\mathcal{F}$ a family of subsets of X such that every element of X appears in at least one subset in $\mathcal{F}$. We say that a subset C of $\mathcal{F}$ is a *set cover* for X if $X = \bigcup_{S \in C} S$ (that is, the union of the sets in C is X). The cardinality of a set cover C is the number of elements in C. (Note that an element of C is a subset of X.) The SET COVER problem is: Given an instance $(X, \mathcal{F})$, and a nonnegative integer k, decide if X has a set cover C in $\mathcal{F}$ of cardinality k. For example, if $X = \{1, 2, 3, 4, 5\}$, $\mathcal{F}= \{\{1\}, \{2\}, \{3\}, \{2, 5\}, \{1, 3, 5\}, \{1, 2, 5\}, \{2, 4, 5\}, \{1, 4, 5\}\}$, and $k = 2$, then the answer is Yes because X has a set cover C in $\mathcal{F}$ of cardinality 2, namely $C = \{\{1, 3, 5\}, \{2, 4, 5\}\}$.

Show that the SET COVER problem is NP-complete. (**Hint.** Reduce from VERTEX COVER.)