CS 5300 01 Advanced Algorithm Design and Analysis
Homework
(Due: 11/30, class time 2:30PM)

(**Total**: 100 points)

1. Let X(1..n) and Y(1..n) contain two lists of n integers, each sorted in nondecreasing order. Give the best (worst-case complexity) algorithm that you can think for finding
(a) the largest integer of all 2n combined elements.
(b) the second largest integer of all 2n combined elements.
(c) the median (or the n^{th} smallest integer) of all 2n combined elements.
For instance, X = (4, 7, 8, 9, 12) and Y = (1, 2, 5, 9, 10), then median = 7, the n^{th} smallest, in the combined list (1, 2, 4, 5, 7, 8, 9, 9, 10, 12). [Hint: use the concept similar to binary search]

2.

1-to-2 PARTITION:

Instance: A finite set of positive integers $Z = \{ z_1 , z_2 , \ldots , z_n \}$.
Question: Is there a subset Z' of Z such that
Sum of all numbers in Z' = 2 x Sum of all numbers in Z-Z'

(a) Obtain the dynamic programming functional equation to solve the 1-to-2 *PARTITION* problem.
(b) Give an algorithm to implement your functional equation.
(c) Give an example of 5 numbers with a total of 21 as an input instance for 1-to-2 *PARTITION* problem, and show how your algorithm works on this input instance.
(d) What is the complexity of your algorithm?

3. Decide *True* or *False* for each of the followings. You MUST briefly justify your answer.

Satisfiability:

Instance: Set U of variables, collection C of clauses over U.
Question: Is there a satisfying truth assignment for C?

(a) If P ≠ NP, then no problem in NP can be solved in polynomial time deterministically.
(b) If a decision problem A is NP-complete, proving that A is reducible to B, in polynomial time, is sufficient to show that B is NP-complete.
(c) It is known that SAT (Satisfiability) is NP-complete, and 3SAT (all clauses have size 3) is NP-complete. 1SAT (all clauses have size 1) is also NP-complete.

4. Given that PARTITION problem (described below) is a NP-Complete problem, prove that the SUM OF SUBSETS problem (described below) is NP-Complete by reducing PARTITION problem to it.

PARTITION:

Instance: A finite set of positive integers $Z = \{ z_1 , z_2 , \dots , z_n \}$.
Question: Is there a subset Z' of Z such that
Sum of all numbers in Z' = Sum of all numbers in Z-Z'

SUM OF SUBSETS:

Instance: A finite set of positive integers $A = \{ a_1, a_2, \dots , a_m \}$ and M.
Question: Is there a subset A' in A s.t. $\sum_{a_i \text{ in } A'} a_i = M$?

(a) Give a nondeterministic polynomial time algorithm for the SUM OF SUBSETS problem.

(b) Define the transformation from the PARTITION problem to the SUM OF SUBSETS problem.
(c) Explain that the transformation described in part (b) satisfies: if the partition problem has a solution then the sum-of-subsets has a solution, and vice versa.

5. Prove that the 0/1 KNAPSACK problem is NP-Hard. (One way to prove this is to prove the decision version of 0/1 KNAPSACK problem is NP-Complete. In this problem, we use PARTITION problem as the source problem.)

(a) Give the decision version of the O/1 KNAPSACK problem, and name it as DK.

(b) Show that DK is NP-complete (by reducing PARTITION problem to DK).

(c) Explain why showing DK, the decision version of the O/1 KNAPSACK problem, is NP-Complete is good enough to show that the O/1 KNAPSACK problem is NP-hard.

6. Optimization PS(3) Problem: Given a set of n program and three storage devices. Let s_i be the amount of storage needed to store the i^{th} program. Let L be the storage capacity of each disk. Determine the maximum number of these n programs that can be stores on the three disks (without splitting a program over the disks).

Use the Approximation PS Algorithm given in the class for the PS(3) problem given above. Show that the following is true.

Let the approximation PS algorithm returns a number C, and let C^* be the optimal (maximum) number of programs that can be stores on the three disks.
(a) Show that the above approximation PS algorithm gives the performance ratio of
$C^* \leq (C + 2)$ OR $C^* / C \leq 1 + 2/C$.
(b) Give an example that achieves the performance ratio of $C^* = (C + 2)$.