COSC 483 – Midterm Exam

Submission and Format Guidelines

Submission: Through Blackboard under a single PDF file. All student names must be printed in the submission. Late submission will not be accepted at all and will result in a grade of 0 (zero).

Format: Your answer to any question should address the following four items

1. A verbal description of your approach or thought process (step-by-step).
2. Solutions: Pseudo code for (i) Brute-force and (ii) Your main approach (e.g., divide and conquer, dynamic programming, etc.)
3. Proof of correctness of your approach
4. Big-O analysis (in detail) of your answers in step #2.

Attention: (Long form) Posting the midterm exam questions and/or receiving help from any Internet medium (e.g., chegg.com) is a violation to the TU Academic Integrity and can result in an F and/or a report to the University.

(Short form) Their answers are 90% incorrect and are very easy to detect. I checked. You'd better answer the questions yourself. You're the one who answer the future interview questions, not them.

Questions #1, 2 and 5 are required. You can choose to answer either question 3 or 4.

1. A position in an array is said to be a k-center if the two values k hops away from it (towards the front and back of the array) are the same. For instance, with $A = [-1,3,1,5,7,3,6,4,-1,9]$ position #4 (value 5) is a 2-center, and position #5 (value 7) is a 4-center. Given an array A of n integers, find the position with the maximal k-center (i.e., the center with maximal radius k) by at least two approaches.
2. Assume that, in Knapsack problem, each weight $W[i]$ can be selected up to a certain $S[i]$ times, e.g., if $W = [2, 5, 8, 13, 17]$ and $S = [3, 2, 2, 1, 1]$ it means weight $W[1] = 2$ can be selected up to $S[1] = 3$ times, weight $W[2] = 5$ can be selected up to $S[2] = 2$ times, etc. Given a capacity C, find the least number of weights that add up closest (or equal) to C. For instance, with $C = 30$ two possible combinations are $[13, 17]$ and $[2, 2, 5, 5, 8, 8]$. $[13, 17]$ is the answer with the least number of weights needed.
3. Given an array $A[]$ of n integers, find two indices s and t such that the difference $(A[s] - A[t])$ is maximized while the sum $(A[s] + A[t])$ is minimized. Note that the integers can be positive or negative
4. Given an array $A[]$ of n elements possibly with duplicates. Design a procedure to remove the duplicates (and retain only the first occurrences in their order) of elements in $A[]$ without sorting.

   Sample: Input $A = [3.1, -2.7, 5.6, 3.1, 6.2, -2.7, -2.7, 3.1, 6.2]$. Output: $[3.1, -2.7, 5.6, 6.2]$.

5. A pirate is standing a the top of a $N \times M$ table where each element in the table represents the number of golden coins he can collect. The rules are simple, the pirate starts at anywhere in the first row, and can only go down either straight, or to the left or to the right (if possible). His goal is to collect as many golden coins as possible. Find a way to help the pirate to do that using dynamic programming.

   For example, with the 4×5 table below

5	4	**10**	8	9
7	11	9	**20**	7
40	13	**11**	6	12
16	27	**100**	20	10

   The best path would be as highlighted.