COM 368: Algorithms II **(Due: 17/03/20)**

Homework #1

Murat Osmanoglu

Homework exercises should be done individually (You should write the solution by yourself). Solutions must be prepared in C language, and submitted electronically as .c file before 11.30 pm on Tuesday March 17. Collaboration is allowed/encouraged on problems, however each student must independently complete their own write-up. No credit will be given to solutions obtained verbatim from the Internet or other sources.

1. Given a set of M positive integers $A = a_1, \ldots, a_M$, develop an dynamic programming that finds a partition A_1, A_2 such that $\mid S_1 - S_2 \mid$ is minimized, where S_i is the sum of elements in A_i.

2. Given bit strings $X = x_1 x_m$, $Y = y_1 y_n$, and $Z = z_1 z_{m+n}$, if Z can be obtained by interleaving the bits in X and Y in a way that it maintains the left-to-right order of the bits in X and Y, then we say Z is an interleaving of X and Y. For example if $X = 101$ and $Y = 01$, then $Z = 10011$ is an interleaving of X and Y, but $Z = 11010$ is not. Develop a (dynamic programming) algorithm that determines whether Z is an interleaving of X and Y or not.

3. Given an array of size n that has the following specifications:

- each element in the array contains either a policeman or a thief
- each policeman can catch only one thief
- policeman cannot catch a thief who is more than K units away from the policeman

Devise a greedy algorithm that finds the maximum number of thieves that can be caught. Also, prove that your algorithm satisfies the greedy-choice property. For the array [P, T, T, P, T] and $k = 1$, your algorithm should output 2 (first policeman catches first thief and second one catches either second or third thief) For the array [T, T, P, P, T, P] and $k = 2$ your algorithm should output 3.

4. Given a set of n activities $A_1, \ldots, A_n$ with the starting and finishing time (s_i, f_i) assigned to each activity and a number of rooms in which activities are performed. Develop a greedy algorithm that assign all the activities to minimum number of rooms so that no two overlapping activities are performed in same room. Also, prove that your algorithm satisfies the greedy-choice property.

Note that, for the last two questions, you don't need to write the whole proof. Just provide the sketch of the proof as 'comments'.