**Design and Analysis of Algorithms**
**Spring 2018**
**Section C, D and E**
**Assignment 3**
**Submission Date: April 10, 2018**

**Problem0**

Read the chapter of dynamic programming from the text book, Introduction to Algorithms, specifically longest common subsequence and matrix chain multiplication problems.

**Problem 1**

Suppose you are managing the construction of billboards on a Highway, a heavily traveled stretch of road that runs west-east for $M$ miles. The possible sites for billboards are given by numbers X= $x1, x2, \ldots, xn$, each in the interval $[0,M]$ (specifying their position along the highway, measured in miles from its western end). If you place a billboard at location $xi$, you receive a revenue of $ri > 0$. Regulations imposed by the county's Highway Department require that no two of the billboards be within less than or equal to 5 miles of each other. You'd like to place billboards at a subset of the sites so as to maximize your total revenue, subject to this restriction. Device an efficient algorithm that takes the input X and compute the sites for billboards that maximize the revenue. Give the pseudo code of your algorithm, also compute its time complexity.

**Problem 2**

Suppose you're managing a consulting team of expert computer hackers, and each week you have to choose a job for them to undertake. Now, as you can well imagine, the set of possible jobs is divided into those that are *low-stress* (e.g., setting up a Web site for a class at the local elementary school) and those that are *high-stress* (e.g., protecting the nation's most valuable secrets). The basic question, each week, is whether to take on a low-stress job or a high-stress job.

If you select a low-stress job for your team in week $i$, then you get a revenue of $li > 0$ dollars; if you select a high-stress job, you get a revenue of $hi > 0$ dollars. The catch, however, is that in order for the team to take on a high-stress job in week $i$, it's required that they do no job (of either type) in week $i - 1$; they need a full week of prep time to get ready for the crushing stress level. On the other hand, it's okay for them to take a lowstress job in week $i$ even if they have done a job (of either type) in week $i - 1$.

So, given a sequence of $n$ weeks, a *plan* is specified by a choice of "low-stress," "high-stress," or "none" for each of the $n$ weeks, with the property that if "high-stress" is chosen for week $i > 1$, then "none" has to be chosen for week $i - 1$. (It's okay to choose a high-stress job in week 1.) The *value* of the plan is determined in the natural way: for each $i$, you add $li$ to the value if you choose "low-stress" in week $i$, and you add $hi$ to the value if you choose "high-stress" in week $i$. (You add 0 if you choose "none" in week $i$.)

Devise an efficient algorithm for given sets of values l1, $l2, \ldots, ln$ and $h1, h2, \ldots, hn$, find a plan of maximum value. Give the pseudo code of your algorithm, also compute its time complexity.

**Problem 3**

Remember the coin changing problem studied in class whose greedy solution does not work for all monetary systems. Now device a dynamic programming solution for the coin changing problem for any monetary system. Given a set of n coins with values {v1, v2, ... , vn} and sum s, compute the minimum number of coins that sum to s. You may assume that there is an infinite collection of coins of all values.

Devise an efficient algorithm for this problem. Give the pseudo code of your algorithm, also compute the time complexity of your algorithm.

**Problem 4**

Suppose you're consulting for a company that manufactures PC equipment and ships it to distributors all over the country. For each of the next *n* weeks, they have a projected *supply si* of equipment (measured in pounds), which has to be shipped by an air freight carrier.

Each week's supply can be carried by one of two air freight companies, A or B.

- Company A charges a fixed rate *r* per pound (so it costs *r* · *si* to ship a week's supply *si*).
- Company B makes contracts for a fixed amount *c* per week, independent of the weight. However, contracts with company B must be made in blocks of four consecutive weeks at a time.

A *schedule*, for the PC company, is a choice of air freight company (A or B) for each of the *n* weeks, with the restriction that company B, whenever it is chosen, must be chosen for blocks of four contiguous weeks at a time. The *cost* of the schedule is the total amount paid to company A and B, according to the description above.

Give an efficient algorithm that takes a sequence of supply values *s*1, *s*2, . . . , *sn* and returns a *schedule* of minimum cost. Give the pseudo code of your algorithm, also compute the time complexity of your algorithm.

***Important Note:***
Do mention your roll number and section on the home work. Your submitted work must be your own contribution. Any sort of plagiarism will be punished severely.