Due date: Email your solutions to cs140_17fall@163.com or submit a written copy to the TA before 7:30 PM, December 21, 2017.

Problem 1 Suppose you have n types of items, where an item of type i has weight w_i and value v_i. You have an infinite number of items of each type, and a knapsack which can hold items of total weight at most W. Describe an algorithm for maximizing the total value of the items you can place in the knapsack, and write down the corresponding DP equation. Your algorithm should run in $O(nW)$ time.

Problem 2 *Multiplication card* is a game played with a row of cards, where each card contains a single positive integer. A player repeatedly takes cards out of the row, and for each card, she gets a number of points equal to the product of the number on the card taken and the numbers on the cards to the left and right of it. The player cannot take the first or last card in the row. The game ends when there are two cards left in the row. The goal of the game is to take cards in an order which minimizes the player's total number of points.

For example, if the cards contain numbers 10 1 50 20 5, and the player takes cards 1, then 20, then 50, she gets a score of

$$10 \times 1 \times 50 + 50 \times 20 \times 5 + 10 \times 50 \times 5 = 8000$$

If she takes cards in the opposite order, i.e. 50, then 20, then 1, the score would be

$$1 \times 50 \times 20 + 1 \times 20 \times 5 + 10 \times 1 \times 5 = 1150$$

Suppose there are n cards in the row. Design an $O(n^3)$ time dynamic programming algorithm to solve the card game, and write down the corresponding DP equation.

Problem 3 Alice loves to eat apples. One day her friend Bob takes her to an apple tree. There are n nodes in the tree, and each node has a certain number of apples. Alice can start at any node of the tree, and can also move to a neighboring node in one step. For each node Alice is at, she eats all the apples there. However, to prevent Alice from eating too many apples, she is only allowed to make k moves in the tree.

Your goal is to help Alice eat as many apples as she can. Design an $O(nk^2)$ time dynamic programming algorithm to solve the problem, and write down the corresponding DP equation.

Problem 4 The *lowest common ancestor (LCA)* of two nodes v and w in a rooted tree is the lowest (i.e. farthest from the root) node that has both v and w as descendants [1]. Suppose we have a tree with n nodes, and m pairs of nodes from the tree $(v_1, w_1), \ldots, (v_m, w_m)$. Design an algorithm to compute the LCAs of all the node pairs in $O((n + m) \log (n))$ time.

[1] We define a node to be a descendant of itself.

Problem 5 Given a sequence of non-negative numbers $a_1, a_2, \ldots, a_n$, let $w(i,j) = \sum_{k=i}^{j} a_k$, for $1 \le i \le j \le n$. Suppose we have a dynamic programming equation

$$f(i,j) = \begin{cases} \min_{i<k\le j}\{f(i,k-1) + f(k,j) + w(i,j)\} & \text{if } i < j \\ 0 & \text{if } i = j \\ +\infty & \text{if } i > j \end{cases}$$

(a) Design an algorithm to compute $f(1,n)$ and give its time complexity.

(b) Given integers $0 < a < b < c < d$, prove that

$$w(a,c) + w(b,d) \le w(b,c) + w(a,d)$$

$$w(b,c) \le w(a,d)$$

(c) Again for integers $0 < a < b < c < d$, prove that

$$f(a,c) + f(b,d) \le f(b,c) + f(a,d)$$

(d) Let $K(i,j) = \arg\min_{i<k\le j}\{f(i,k-1) + f(k,j) + w(i,j)\}$, for $1 \le i < j \le n$. Prove that

$$K(i,j-1) \le K(i,j) \le K(i+1,j)$$

(e) Use the inequalities above to design a fast way to compute $f(1,n)$. Give the time complexity of your method.

The End.