COMP 3711 – Design and Analysis of Algorithms
2017 Fall Semester – Written Assignment # 3
Distributed: Oct 25 2017 – Due: Nov 10, 2017

Your solutions should contain (i) your name, (ii) your student ID #, and (iii) your email address

Some Notes:

- Please write the solutions in order with A4 papers.
- Please write clearly and briefly.
- Please follow the guidelines on doing your own work and avoiding plagiarism given on the class home page.
  In particular ***don't forget to acknowledge individuals who assisted you, or sources where you found solutions.*** Failure to do so will be considered plagiarism.
- Please make a *copy* of your assignment before submitting it. If we can't find your answers, we will ask you to resubmit the copy.
- The default base for logarithms will be 2, i.e., $\log n$ will mean $\log_2 n$. If another base is intended, it will be explicitly stated, e.g., $\log_3 n$.
- Each problem is worth 25 points.
- In addition to submitting a hard copy, you will be asked to submit a PDF online for our records. This PDF can be generated by Latex, from Word or a scan of a (legible) handwritten solution.
- If you have any questions about the assignment, contact hwuau@cse.ust.hk for more information.

Problem 1: Given a set $S = \{a_1, a_2, \cdots, a_n\}$ of n positive integers and a positive integer W, design an $O(nW)$-time dynamic programming algorithm to decide whether there is a subset $K \subseteq S$ such that the sum of all numbers in K is W. For example, if $S = \{1, 4, 7, 3, 5\}$ and $W = 11$, then the answer is "yes" since K can be $\{4, 7\}$. (Your algorithm doesn't have to output the subset K.) If no such subset exists, your algorithm should return "no".

Problem 2 In this question, you are required to solve the 0-1 Knapsack problem for *two* knapsacks.

You are given a set of n objects. The weights of the objects are $w_1, w_2, \ldots, w_n$, and the values of the objects are $v_1, v_2, \ldots, v_n$. You are given two knapsacks each of weight capacity C. If an object is taken, it may placed in one knapsack or the other, but not both. All weights and values are positive integers.

Design an $O(nC^2)$ dynamic programming algorithm that determines the maximum value of objects that can be placed into the two knapsacks. Your algorithm should also determine the contents of each knapsack. Justify the correctness and running time of your algorithm.

Problem 3 Given two strings $x[1..n]$ and $y[1..m]$, we would like to transform x to y by the following operations:

Insert a character into x (at any position).
Delete a character from x (at any position).
Replace a character in x by another character.

Example: $x = abcd$, $y = bcfe$, one possible way to transform x to y is

1. Delete the beginning a from x. x becomes bcd.
2. Replace character d in x by character f. x becomes bcf.
3. Insert character e at the end of x. x becomes $bcfe$.

Here is another possible way

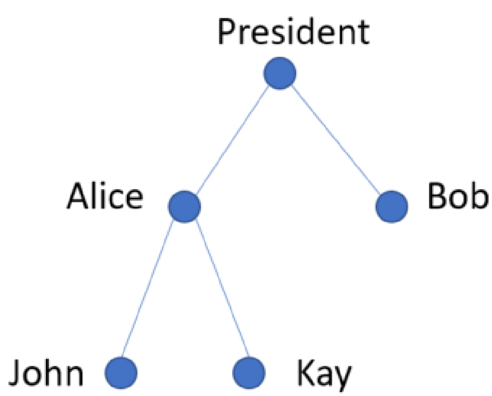
1. Delete the beginning a from x. x becomes bcd.
2. Insert character f before character d in x. x becomes $bcfd$.
3. Replace character d in x by character e. x becomes $bcfe$.

Design an $O(mn)$ algorithm that returns the minimum number of operations we need to transform x to y.

Problem 4 Professor Smart is consulting for the president of a corporation that is planning a company party. The company has a hierarchical structure; that is, the supervisor relation forms a tree rooted at the president. The personnel office has ranked each employee with a conviviality rating, which is a real number. In order to make the party fun for all attendees, the president does not want both an employee and his or her immediate supervisor to attend.

Professor Smart is given the tree $T =< V, E >$ that describes the structure of the corporation as well as the conviviality ratings of employees (including the president). Design an $O(|V|)$ ($|V|$ is the size of V) algorithm to make up a valid guest list that maximizes the sum of the conviviality ratings of the guests (a guest list is valid if: 1. it includes the president 2. an employee and his or her immediate supervisor can not appear in the list simultaneously)

Example:



Conviviality Ratings

President	3
Alice	4
Bob	2
John	5
Kay	1

Some possible guest lists:

1. {John, President}, valid, sum of conviviality ratings: $5 + 3 = 9$
2. {John, Kay, President}, valid, sum of conviviality ratings: $5 + 1 + 3 = 9$
3. {Alice, Bob }, not valid, the president must be in the guest list
4. {John, Kay, President, Bob}, not valid, Bob's immediate supervisor (i.e., the president) is in the list.