Due Date: March 26, 2013

Problem 1: Let $G = (V, E)$ be an undirected graph with nonnegative edge weights. Suppose we have computed a minimum spanning tree (MST) T of G, and we have also computed shortest paths to all nodes from a particular node $s \in V$. Now, let the weight of each edge be increased by 1.

i) Does the MST change? Give an example where it changes or prove that it cannot change.

ii) Do the shortest paths change? Give an example where they change, or prove they cannot change.

Problem 2: A palindrome is any string that is exactly the same as its reversal, such as A, or NITIN, or MALAYALAM.

i) Design an efficient dynamic programming algorithm to find the length of the longest *subsequence* of a given string, that is also a palindrome. For e.g., the longest palindrome subsequence of BATMANISAFICTIONALCHARACTER is AANISINAA, so given that string as input, your algorithm should return 9.

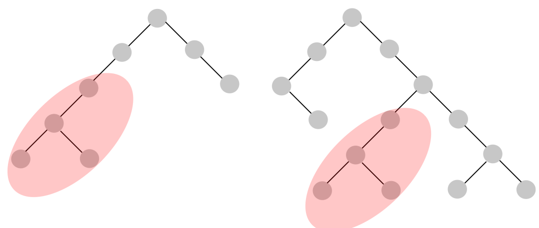
ii) Any string can be decomposed into a sequence of palindromes. For e.g., the string BUBBASEESABANANA can be broken into palindromes in the following ways (and many others):

   1. BUB + BASEESAB + ANANA
   2. B + U + BB + A + SEES + ABA + NAN + A
   3. B + U + B + B + A + S + E + E + S + A + B + A + N + A + N + A

   Design an efficient dynamic programming algorithm to find the smallest number of palindromes that make up a given input string. For e.g., given the above input string, your algorithm would return 3.

Problem 3: You are given a steel chain made of n links. You can split the chain into 2 pieces, at any location. The cost of splitting a chain, is the total length of the piece, n in this case, regardless of where you split it. Now suppose you want to break the chain into many pieces. The order in which you break the chain can affect the total cost involved in breaking it. For e.g., if you want to break a chain made of 30 links at positions 5 and 17, then making a cut at position 5 first gives a total cost of 30 + 25 = 55, while making a cut at position 17 first gives a total cost of 30 + 17 = 47. Give an efficient dynamic programming algorithm, that given m positions in a chain, finds the minimum cost of cutting the chain into $m + 1$ pieces. (**Hint:** *This problem is similar to the matrix chain multiplication problem.*)

Problem 4: A subtree of a (rooted, ordered) binary tree T consists of a node and all its descendants. Design an efficient algorithm to compute the *largest common subtree* of two given binary trees T_1 and T_2. The largest common subtree is the the largest subtree of T_1 that is identical to a subtree in T_2. The contents of the nodes do not play a role, only the underlying combinatorial structure. For e.g., in the figure below, the largest common subtrees have been shaded.



Problem 5: You are given n integers $x_1, x_2, ..., x_n$. Let S be the sum of all the integers, i.e. $S = \sum_{i=1}^{n} x_i$. You want to determine whether it is possible to partition $\{1, ..., n\}$ into 3 disjoint subsets A,B,C such that

$$\sum\nolimits_{x_i \in A} x_i = \sum\nolimits_{x_j \in B} x_j = \sum\nolimits_{x_k \in C} x_k = \frac{S}{3} \tag{1}$$

For e.g., for input $(5, 2, 1, 6, 2, 4, 7)$, the answer is yes, because you can partition it into $(2, 7)$,$(6, 1, 2)$,$(4, 5)$. However, for the input $(2, 2, 4, 4, 5, 7)$, the answer is no. Given an efficient dynamic programming algorithm for the above set partition problem that runs in polynomial time in n and in S.