Shri G. S. Institute of Technology and Science
Department Of Computer Engineering
CO 3406: Design and Analysis of Algorithms
Lab Assignment # 05 (Dynamic Programming Algorithms)
Marks: 10 points

Submission Date: 19 March 2017@23:59
Demo Date: 20 March 2017 - 25 March 2017
Late Submission: Not allowed
No copying allowed. If found then students involved in copying will fail in this course.

Q. 32. Given 3 strings of all having length < 100, write a program to find the longest common sub-sequence in all three given sequences.

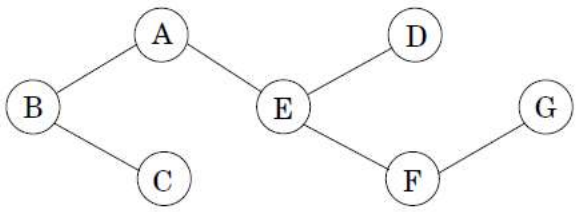
Q. 33. Given a sequence of matrices, find the most efficient way to multiply these matrices together. The problem is not actually to perform the multiplications, but merely to decide in which order to perform the multiplications.

Write a program to find the contiguous subsequence of maximum sum (a subsequence of length zero has sum zero). A contiguous subsequence of a list S is a subsequence made up of consecutive elements of S. For instance, if S is 5; 15; 30; 10; 5; 40; 10; then 15; 30; 10 is a contiguous subsequence but 5; 15; 40 is not. For the preceding example, the answer would be 10; 5; 40; 10, with a sum of 55. Give a linear-time algorithm for the following task:

Q. 34. A subsequence is palindromic if it is the same whether read left to right or right to left. For instance, the sequence A;C; G; T; G; T;C; A; A; A; A; T;C;G has many palindromic subsequences, including A;C; G;C;A and A; A; A;A (on the other hand, the subsequence A;C; T is not palindromic). Devise an algorithm that takes a sequence x[1 : : : n] and returns the (length of the) longest palindromic subsequence. Its running time should be $O(n^2)$.

Q. 35. A list of n positive integers a1; a2; : : : ; an; and a positive integer t is given. Write a program to find subset of the ai's add up to t? (You can use each ai at most once.)

Q. 36. A vertex cover of a graph G = (V;E) is a subset of vertices S ⊆ V that includes at least one endpoint of every edge in E. Give & write a linear-time algorithm for finding the size of the smallest vertex cover of T. For instance, in the following tree, possible vertex covers include {A;B;C;D;E; F;G} and {A;C;D; F} but not {C;E;F}. The smallest vertex cover has size 3: {B;E;G}.



Q. 37. Write a program for the following 3-PARTITION problem. Given integers a1; : : : ; an, we want to determine whether it is possible to partition of {1; : : : ; n} into three disjoint subsets I;J;K such that

$$\sum_{i \in I} a_i = \sum_{j \in J} a_j = \sum_{k \in K} a_k = \frac{1}{3} \sum_{i=1}^{n} a_i$$

For example, for input (1; 2; 3; 4; 4; 5; 8) the answer is yes, because there is the partition (1; 8), (4; 5), (2; 3; 4). On the other hand, for input (2; 2; 3; 5) the answer is no. Write a program and analyze a dynamic programming algorithm for 3-PARTITION that runs in time polynomial in n.

Q. 38. Write a program that calculates the highest sum of numbers passed on a route that starts at the top and ends somewhere on the base.

```
          7
        3   8
      8   1   0
    2   7   4   4
  4   5   2   6   5
```

(Figure 1)

For the above figure shows a number triangle and its output is 30(7,3,8,7,5). Each step can go either diagonally down to the left or diagonally down to the right.

Q. 39. Cutting cloth: You are given a rectangular piece of cloth with dimensions X x Y , where X and Y are positive integers, and a list of n products that can be made using the cloth. For each product i 2 [1; n] you know that a rectangle of cloth of dimensions ai x bi is needed and that the final selling price of the product is ci. Assume the ai, bi, and ci are all positive integers. You have a machine that can cut any rectangular piece of cloth into two pieces either horizontally or vertically. Design an algorithm that determines the best return on the X xY piece of cloth, that is, a strategy for cutting the cloth so that the products made from the resulting pieces give the maximum sum of selling prices. You are free to make as many copies of a given product as you wish, or none if desired.

Q. 40. Write a program using dynamic programming for yours own two problems and prove its complexity in polynomial.