KARPAGAM COLLEGE OF ENGINEERING

17PE05/17FE05/17ME33/17LE33/17TE33/17EE33/17NE33
DESIGN AND ANALYSIS OF ALGORITHMS

SESSION 3.2

1. 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. There are many options to multiply a chain of matrices because matrix multiplication is associative i.e. no matter how one parenthesize the product, the result will be the same.

Example:

if you had four matrices A, B, C, and D, you would have:

(ABC)D = (AB)(CD) = A(BCD) = ...

However, the order in which one parenthesize the product affects the number of simple arithmetic operations needed to compute the product, or the efficiency.

For example:

A: 10 × 30 matrix

B : 30 × 5 matrix

C : 5 × 60 matrix

Then,

(AB)C = (10×30×5) + (10×5×60)

= 1500 + 3000

= 4500 operations

A(BC) = (30×5×60) + (10×30×60)

= 9000 + 18000

= 27000 operations.

Given an array **arr[]** which represents the chain of matrices such that the ith matrix Ai is of dimension **arr[i-1] x arr[i]**. Your task is to write a function that should print the minimum number of multiplications needed to multiply the chain.

Input: p[] = {40, 20, 30, 10, 30}
Output: 26000
There are 4 matrices of dimensions 40x20,
20x30, 30x10 and 10x30. Let the input 4
matrices be A, B, C and D. The minimum
number of multiplications are obtained
by putting parenthesis in following way
(A(BC))D --> 20*30*10 + 40*20*10 + 40*10*30

Input: p[] = {10, 20, 30, 40, 30}
Output: 30000
There are 4 matrices of dimensions 10x20,
20x30, 30x40 and 40x30. Let the input 4
matrices be A, B, C and D. The minimum
number of multiplications are obtained by
putting parenthesis in following way
((AB)C)D --> 10*20*30 + 10*30*40 + 10*40*30

Input:

The first line of the input contains an integer **T**, denoting the number of test cases. Then **T** test case follows. The first line of each test case contains an integer **N**, denoting the number of elements in the array.

	Then next line contains **N** space separated integers denoting the values of the element in the array. **Output:** For each test case the print the minimum number of operations needed to multiply the chain. **Constraints:** 1<=T<=100 2<=N<=100 1<=A[]<=500 **Example:** **Input:** 2 5 1 2 3 4 5 3 3 3 3 **Output:** 38 27
2.	Given a String, find the longest palindromic subsequence **Input:** The first line of input contains an integer T, denoting no of test cases. The only line of each test case consists of a string S(only lowercase) **Output:** Print the Maximum length possible for palindromic subsequence. **Constraints:** 1<=T<=100 1<=|Length of String|<=1000 **Examples:** **Input:** 2 bbabcbcab abbaab **Output:** 7 4
3.	Given two sequences, find the length of longest subsequence present in both of them. Both the strings are of uppercase. **Input:** First line of the input contains no of test cases **T**,the **T** test cases follow. Each test case consist of 2 space separated integers **A** and **B** denoting the size of string **str1** and **str2** respectively. The next two lines contains the 2 string **str1** and **str2** . **Output:** For each test case print the length of longest common subsequence of the two strings . **Constraints:** 1<=T<=200 1<=size(str1),size(str2)<=100 **Example:** **Input:** 2 6 6

	ABCDGH AEDFHR 3 2 ABC AC **Output:** 3 2 **Explanation** LCS for input Sequences “ABCDGH” and “AEDFHR” is “ADH” of length 3. LCS of "ABC" and "AC" is "AC" of length 2