

ASK/VIEW DOUBT

SOLUTION

HINT

Problem

Result

Matrix Chain Multiplication

Send Feedback

Given a chain of matrices A1, A2, A3,.....An, you have to figure out the most efficient way to multiply these matrices i.e. determine where to place parentheses to minimise the number of multiplications.

You will be given an array p[] of size n + 1. Dimension of matrix Ai is p[i - 1]*p[i]. You need to find minimum number of multiplications needed to multiply the chain.

Input Format :

Line 1 : Integer n i.e. number of matrices
Line 2 : n + 1 integers i.e. elements of array p[]

Output Format :

Line 1 : Minimum number of multiplication needed

Constraints :

1 <= n <= 100

Sample Input 1 :

3
10 15 20 25

Sample Output :

8000

Sample Output Explanation :

There are two ways to multiply the chain - A1*(A2*A3) or (A1*A2)*A3.

If multiply in order A1*(A2*A3) then number of multiplications required are 15000.

If multiply in order (A1*A2)*A3 then number of multiplications required are 8000.

Thus minimum number of multiplications required are 8000

1#include<bits/stdc++.h>

2// int help(int* p, int si, int ei,int**out){

3// if(si==ei || si==ei-1){

4// return 0;

5// }

6

7// if(out[si][ei]!=-1){

8// return out[si][ei];

9// }

10

11// for(int k = si+1; k<=ei-1;k++){

12// int a = help(p,si,k,out) + help(p,k,ei,out) + p[si]*p[k]*p[ei];

13// if(a<min){

14// min=a;

15// }

16// }

17// out[si][ei] = min;

18// return min;

19

20// }

21

22

23int mcm(int* p, int n){

24

25/* Don't write main().

26* Don't read input, it is passed as function argument.

27* Return output and don't print it.

28* Taking input and printing output is handled automatically.

29*/

30int out[n+1][n+1];

31

32for(int i=n;i>=0;i--){

33for(int j=0;j<=n;j++){

34if(i==j || i==j-1){

35out[i][j] = 0;

36continue;

37}

38int min = INT_MAX;

PREVIOUS

NEXT

CUSTOM INPUT

SUBMIT SOLUTION