

Name	RUPALI SAWALE
UID no.	2021700056
Experiment No.	5

AIM:	Dynamic Programming - Matrix Chain Multiplication.
Program	
PROBLEM STATEMENT:	Use Dynamic Programming method to find the optimal way to multiply(parenthesize) the matrices to find the minimum number of multiplications required to solve the matrix.
ALGORITHM/ THEORY:	<p>Matrix chain multiplication :</p> <p>Is an optimization problem concerning the most efficient way to multiply a given sequence of matrices. The problem is not actually to <i>perform</i> the multiplications, but merely to decide the sequence of the matrix multiplications involved. The problem may be solved using dynamic programming.</p> <p>There are many options because matrix multiplication is associative. In other words, no matter how the product is parenthesized, the result obtained will remain the same. For example, for four matrices <i>A</i>, <i>B</i>, <i>C</i>, and <i>D</i>, there are five possible options:</p> $((AB)C)D = (A(BC))D = (AB)(CD) = A((BC)D) = A(B(CD)).$
PROGRAM:	<pre>#include<stdio.h> #include<time.h> int mat[100][100],s[100][100],count=0; int MCM(int p[], int i, int j){ if(i==j){ mat[i][j] = 0; return 0; } mat[i][j] = 30000; for(int k=i; k<j; k++){ count = MCM(p,i,k) + MCM(p,k+1,j) + p[i-1]*p[k]*p[j]; } }</pre>

```

        if(count<mat[i][j]){
            mat[i][j] = count;
            s[i][j] = k;
        }
    }
    return mat[i][j];
}

void POP(int i,int j){
    if(i==j)
        printf("S%d",i);
    else{
        printf("(");
        POP(i,s[i][j]);
        POP(s[i][j]+1,j);
        printf(")");
    }
}

void main(){
    int num;
    printf("\n The number of inputs : ");
    scanf("%d",&num);
    int p[num];
    // printf("\nEnter the order of matrices: ");
    for(int i=0;i<num;i++){
        printf("\n The value for place is %d: ",i+1);
        scanf("%d",&p[i]);
    }
    printf("\n Minimum number of multiplications : %d\n\n",MCM(p,1,num-1));
    for(int i=1;i<num;i++){
        for(int j=1;j<num;j++){
            printf("%d\t",mat[i][j]);
        }
        printf("\n");
    }
    printf("\n The optimal solution : \n");
    POP(1,num-1);
}

```

RESULT:

✓ ↗ 📄

The number of inputs : 4

The value for place is 1: 10

The value for place is 2: 20

The value for place is 3: 30

The value for place is 4: 40

< Minimum number of multiplications : 18000

0	6000	18000
0	0	24000
0	0	0

The optimal solution :
((S1S2)S3)

...Program finished with exit code 0
Press ENTER to exit console. □

CONCLUSION:	The Dynamic Programming steps was used to solve the Matrix Chain Multiplication problem.
--------------------	--