|  |  |
| --- | --- |
| **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:** | **Matrix chain multiplication** :  Is an [optimization problem](https://en.wikipedia.org/wiki/Optimization_problem) concerning the most efficient way to [multiply](https://en.wikipedia.org/wiki/Matrix_multiplication) a given sequence of [matrices](https://en.wikipedia.org/wiki/Matrix_(mathematics)). The problem is not actually to *perform* the multiplications, but merely to decide the sequence of the matrix multiplications involved. The problem may be solved using [dynamic programming](https://en.wikipedia.org/wiki/Dynamic_programming).  There are many options because matrix multiplication is [associative](https://en.wikipedia.org/wiki/Associativity). In other words, no matter how the product is [parenthesized](https://en.wikipedia.org/wiki/Bracket_(mathematics)), the result obtained will remain the same. For example, for four matrices *A*, *B*, *C*, and *D*, there are five possible options:  ((*AB*)*C*)*D* = (*A*(*BC*))*D* = (*AB*)(*CD*) = *A*((*BC*)*D*) = *A*(*B*(*CD*)). |
| **PROGRAM:** | #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];  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:** | |
| **CONCLUSION:** | The Dynamic Programming steps was used to solve the Matrix Chain Multiplication problem. |