|  |  |
| --- | --- |
| **NAME:** | Hardik Garg |
| **UID:** | 2021300036 |
| **SUBJECT** | Design and Analysis of Algorithms |
| **EXPERIMENT NO:** | 4 |
| **AIM:** | To implement Matrix Chain Multiplication |
| **Algorithm:** | **MATRIX-CHAIN-ORDER (p)**  1. n length[p]-1  2. for i ← 1 to n  3. do m [i, i] ← 0  4. for l ← 2 to n // l is the chain length  5. do for i ← 1 to n-l + 1  6. do j ← i+ l -1  7. m[i,j] ← ∞  8. for k ← i to j-1  9. do q ← m [i, k] + m [k + 1, j] + pi-1 pk pj  10.If q < m [i,j]  11.then m [i,j] ← q  12.s [i,j] ← k  13.return m and s.  **PRINT-OPTIMAL-PARENS (s, i, j)**  1. if i=j  2. then print "A"  3. else print "("  4. PRINT-OPTIMAL-PARENS (s, i, s [i, j])  5. PRINT-OPTIMAL-PARENS (s, s [i, j] + 1, j)  6. print ")" |
| **Code:** | #include<bits/stdc++.h>  using namespace std;  int main()  {  int n;  cout<<"Enter number of dimensions:";  cin>>n;  int a[n];  cout<<"Enter dimensions:";  for(int i=0;i<n;i++)  cin>>a[i];  int m[n][n];  for(int i=0;i<n;i++)  {  for(int j=0;j<n;j++)  {  if(i==0 || j==0 || i==j)  m[i][j]=0;  }  }  for(int r=1;r<n;r++)  {  int t,j=1+r;  for(int i=1;j<n;i++,j=i+r)  {  for(int k=i;k<j;k++)  {  t=m[i][k]+m[k+1][j]+a[i-1]\*a[k]\*a[j];  if(k==i)  m[i][j]=t;  else  m[i][j]=min(t,m[i][j]);  }  // cout<<"m["<<i<<"]["<<j<<"]="<<m[i][j]<<endl;  }  }  cout<<"Optimal cost for parenthesization is:"<<m[1][n-1]<<endl;  } |
| **Output** |  |
| **Conclusion:** | Thus, after performing this experiment I understood how matrix chain multiplication works and how significant it is while multiplying metrices |