DAA Lab-10

Name: Kshitij Kumar Sharma Roll No.: 1905514 Date: 08/10/2021

**Q-10.1)** Write a program to solve the matrix chain multiplication problem following dynamic programming approach.

**Program:**

/\*

Written by: Kshitij Kumar Sharma Roll No.: 1905514

Idea of the solution:

Matrix chain multiplication is an optimization problem concerning the most efficient way to multiply a given sequence of matrices. The problem is not actually to perform the multiplications, but merely to decide the sequence of the matrix multiplications involved. The problem is solved using dynamic programming as it can be divided into shared sub problems.

\*/

#include <bits/stdc++.h>

using namespace std;

int dp[100][100],a[100][100];

void optimalParents(int i,int j) //Prints optimal parents

{

if(i==j)

cout<<"A"<<i;

else

{

cout<<"(";

optimalParents(i,a[i][j]);

optimalParents(a[i][j]+1,j);

cout<<")";

}

}

int matrixChain(int\* p, int i, int j) //Calculates matrix chain multiplication

{

int q;

if(i==j)

return 0;

if (dp[i][j] != -1)

return dp[i][j];

dp[i][j]=INT\_MAX;

for (int k=i;k<j;k++)

{

q=matrixChain(p,i,k)+matrixChain(p,k+1,j)+p[i-1]\*p[k]\*p[j];

if(q<dp[i][j])

{

dp[i][j]=q;

a[i][j]=k;

}

}

cout<<dp[i][j]<<endl;

return dp[i][j];

}

int MatrixChainOrder(int\* p, int n) //calls matricChain()

{

int i=1,j=n-1;

return matrixChain(p,i,j);

}

int main() //Driver code

{

int n,i;

cin>>n;

int arr[n];

for(i=0;i<n;i++)

cin>>arr[i];

memset(dp, -1, sizeof dp);

memset(a, -1, sizeof a);

int s=MatrixChainOrder(arr, n);

cout<<endl;

optimalParents(1,n-1);

cout<<endl;

cout << "Minimum number of multiplications is "<< s<<endl;

}

**Output:**

