ADA External Viva

- Rubleen Kaur
07613203118

Ans 1.

```
ALGORITHM : MATRIX - CHAIN - ORDER ( p, n)
{
    for i := 1 to n do
        m [i,i] := 0;
    for L := 2 to n do // L is the chain length
    {
        for i := 1 to n-L+1 do
            j = i+L-1;
            m [i,j] := ∞;
            for (k := i to j-1 do
            {
                q := m[i,k] + m[k+1,j] + p[i-1]p[k]p[j]
                if q < m [i,j] then
                {
                    m[i,j] := q;
                }
            }
            s[i,j] := k;
    }
    write (m[1,n], s[1,n]);
}
```

The time complexity of the algo is $O(n^3)$.

↑ worst case.

Code :

```
# include <stdio.h>
# include <limits.h>
# include <time.h>
# include <process.h>
# include <unistd.h>

int MatrixChainMultiplication (int p[], int n)
{
    int m[n][n];
    int i, j, k, l, q;
    for (i=0; i<n; i++)
        m[i][i] = 0;
    for (l=2; l<n; l++)
    {
        for (i=1; i<n-l+1; i++)
        {
            j = i+l-1;
            m[i][j] = INT_MAX;
            for (k=0; k<=j-i; k++)
            {
                q = m[i][k] + m[k+1][j] + p[i-1]*p[k]*p[j];
                if (q<m[i][j])
                {
                    m[i][j] = q;
                }
            }
        }
        sleep(5);
    }
    return m[1][n-1];
}
```

```
int main()
{
    int n, i;
    float start, end, timecomplexity;
    printf("\n Rubleen Kaur - 07613203118");
    printf("\n Enter number of matrices : ");
    scanf("%d", &n);
    n++;
    int arr[n];
    printf("Enter dimensions \n");
    for(i=0; i<n; i++)
    {  printf("Enter d %d : ", i);
       scanf("%d", &arr[i]);
    }
    int size = sizeof(arr) / sizeof(arr[0]);
    start = clock();
    timeComplexity = (end - start) / CLK_TCK;
    printf("Time Complexity : %f", timeComplexity);
    return 0;
}
```

Output :-

OUTPUT :

```
Rubleen Kaur - 07613203118
Enter number of matrices : 12
Enter Dimensions
Enter d0 : 3
Enter d1 : 4
Enter d2 : 5
Enter d3 : 6
Enter d4 : 9
Enter d5 : 12
Enter d6 : 4
Enter d7 : 6
Enter d8 : 9
Enter d9 : 12
Enter d10 : 11
Enter d11 : 34
Enter d12 : 65

Minimum number of multiplications : 9690
Time Complexity : 55.070999.
```

Ans 2 The all-pair shortest path algorithm is also known as Floyd-Warshall algorithm is used to find all pair shortest path problem from a given weighted graph. As a result of this algorithm, it will generate a matrix, which will represent the minimum distance from any node to all other nodes in the graph.

Algorithm:

floydWarshal (cost)

I/P: Cost matrix of given Graph
O/P: Matrix for shortest path b/w any vertex to other vertex.

```
Begin
 for k := 0 to n, do
    for i := 0 to n, do
       for j := 0 to n, do

          if (cost[i,k] + cost[k,j] < cost[i,j]), then
            cost[i,j] := cost[i,k] + cost[k,j]
          done
       done
    done
    display cost matrix
End
```