



# Design & Analysis Of Algorithm

## Lab Experiment -10

NAME : SURIYAPRAKASH.C

ROLL NO:CH.EN.U4CSE20170

SUBJECT: DESIGN & ANALYSIS OF ALGORITHM

SUBJECT CODE: 19CSE302

Submitted to – Mrs. Ashwini,  
Department of CSE,  
ASE Chennai campus.

### AIM:

To write an algorithm to implement Matrix Chain Multiplication.

### ALGORITHM:

- 1) Begin
- 2) define table minMul of size  $n \times n$ , initially fill with all 0s
- 3) for length := 2 to  $n$ , do
- 4) for  $i := 1$  to  $n - \text{length}$ , do
- 5)  $j := i + \text{length} - 1$
- 6)  $\text{minMul}[i, j] := \infty$
- 7) for  $k := i$  to  $j - 1$ , do
- 8)  $q := \text{minMul}[i, k] + \text{minMul}[k + 1, j] + \text{array}[i - 1] * \text{array}[k] * \text{array}[j]$
- 9) if  $q < \text{minMul}[i, j]$ , then  $\text{minMul}[i, j] := q$
- 10) done
- done
- done
- 11) return  $\text{minMul}[1, n - 1]$
- 12) End

### CODE SCREEN:

```
import sys
def MatrixChainOrder(p, i, j):
    if i == j:
        return 0

    _min = sys.maxsize
    for k in range(i, j):

        count = (MatrixChainOrder(p, i, k)
                 + MatrixChainOrder(p, k + 1, j)
                 + p[i-1] * p[k] * p[j])
```

```
        if count < _min:
            _min = count
    return _min

if __name__ == '__main__':
    arr = [1,3,4,5,6]
    N = len(arr)

    print("Minimum number of multiplications is ",
          MatrixChainOrder(arr, 1, N-1))
```

#### OUTPUT SCREEN :

```
PS D:\python> & C:/Users/HP/AppData/Local/Programs/Python/Python310/python.exe d:/python/DAA/matrixMultiplication.py
Minimum number of multiplications is 62
PS D:\python> 
```

#### TIME COMPLEXITY:

$O(n^3)$

#### SPACE COMPLEXITY:

$O(n^2)$

#### RESULT:

I have studied and understood the matrix chain multiplication program in python language and executed the program successfully.

THANK YOU !!