Code for Matrix Chain Multiplication Algorithm

import java.util.\*;

public class EXPT\_07\_MatrixChainMultiplication {

    // Matrix Arr[i] has dimension p[i-1] \* p[i] for i = 1, 2, ..., n

    // For example in given matrix <5, 8, 10, 3, 6, 11, 2>, p[0] = 5, p[1] = 8 and

    // so on...

    static int Matrix\_Computation(int p[], int n) {

        int m[][] = new int[n][n];

        int i, j, k, L, q = 0;

        // L is the chain length

        // m[i, j] = Minumum number of scalar multiplications needed to compute the

        // matrix

        // A[i]A[i+1]...A[j] = A[i..j] where dimension of A[i] is p[i-1]\*p[i]

        // cost is 0 when multiplying one matrix

        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;

                if (j == n)

                    continue;

                m[i][j] = Integer.MAX\_VALUE;

                for (k = i; k <= j - 1; k++) {

                    // q is the cost per scalar multiplication

                    q = m[i][k] + m[k + 1][j] + p[i - 1] \* p[k] \* p[j];

                    if (q < m[i][j])

                        m[i][j] = q;

                }

            }

        }

        for (L = 2; L < n; L++) {

            for (i = 1; i < n - L + 1; i++) {

                j = i + L - 1;

                System.out.print("\nm[" + i + "][" + j + "]: " + m[i][j]);

            }

        }

        return m[1][n - 1];

    }

    public static void main(String args[]) {

        Scanner in = new Scanner(System.in);

        int num, n\_o\_m;

        System.out.print("Enter the number of values in the matrix: ");

        num = in.nextInt();

        int arr[] = new int[num];

        for (int i = 0; i < num; i++) {

            System.out.print("\nEnter value " + (i + 1) + ": ");

            arr[i] = in.nextInt();

        }

        long start = System.currentTimeMillis();

        n\_o\_m = Matrix\_Computation(arr, num);

        long end = System.currentTimeMillis();

        System.out.print("\nThe minimum number of multiplications is " + n\_o\_m);

        System.out.print("\nThe time taken by Matrix Chain Multiplication Algorithm using Dynamic Programming is "

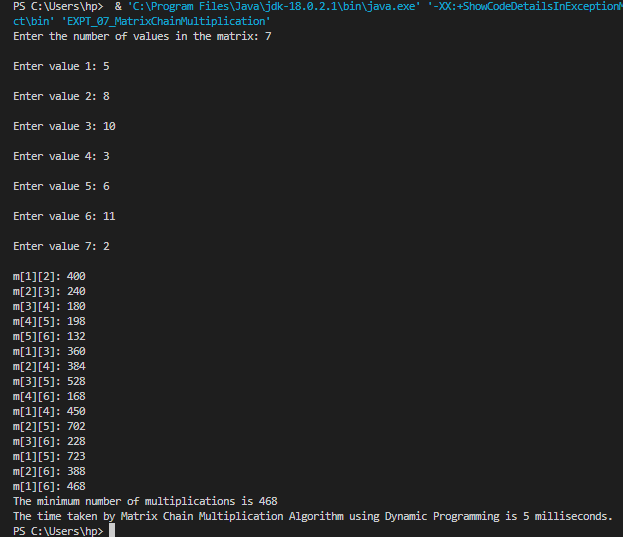
                + (end - start) + " milliseconds.");

    }

}

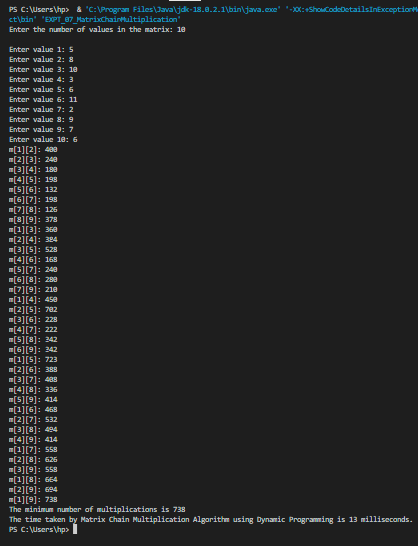
Case 1: When the number of values is 7

Time taken is 5 milliseconds.



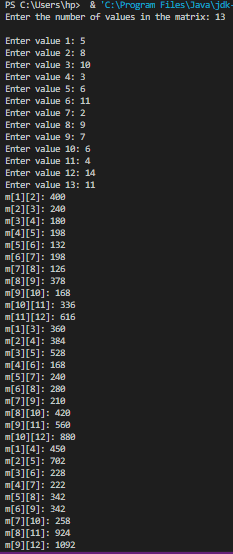
Case 2: When the number of values is 10

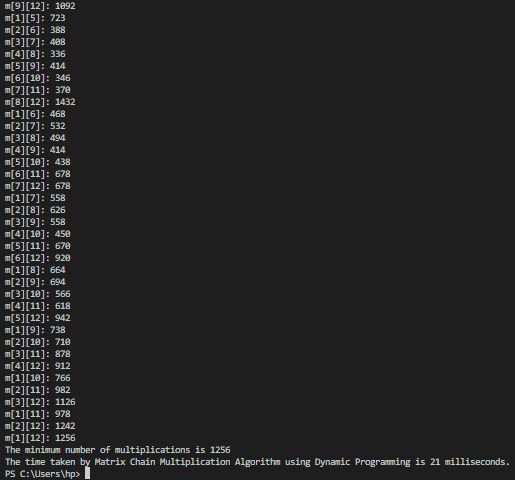
Time taken is 13 milliseconds.



Case 3: When the number of values is 13

Time taken is 21 milliseconds.





Case 4: When the number of values is 16

Time taken is 39 milliseconds.

