

Which of the following is used to solve the Matrix Chain Multiplication problem efficiently?

☐ Divide and Conquer

☒ Dynamic Programming



☐ Greedy Algorithm

☐ Backtracking

What is the time complexity of the Matrix Chain Multiplication algorithm using dynamic programming?

☐  $O(n^4)$

☒  $O(n^3)$  ✓

☐  $O(n^2)$

☐  $O(n \log n)$

Status **Correct**

Mark obtained **1/1**

Hints used **0**

Level **Easy**

Question type **MCQ Single Correct**

Subject **Algorithms**

Topic **Dyn**

Blooms taxonomy **Understand**

☐ Show solution

What is the minimum number of matrices required for Matrix Chain Multiplication to make sense?

☐ 1

☐ 4

☐ 3

☒ 2



## Multi Choice Type Question

If there are  $n$  matrices, what is the size of the dimensions array used in Matrix Chain Multiplication?

☐  $2n$

☐  $n-1$

☐  $n$

☒  $n+1$  ✓

Status	<b>Correct</b>	Mark obtained	<b>1/1</b>	Hints used	<b>0</b>	Level	<b>Easy</b>	Question type	<b>MCQ Single Correct</b>	Subject	<b>Algorithms</b>	Topic	<b>Dynamic Programming</b>	Sub Topic	<b>Dynamic Programming</b>
Blooms taxonomy	<b>Understand</b>														

☐ Show solution

In Matrix Chain Multiplication, which of the following represents the recursive subproblem?

☒ Finding the minimum number of scalar multiplications for a matrix chain



☐ Storing the matrix dimensions

☐ Multiplying matrices from index  $i$  to  $j$

☐ Calculating the matrix product of matrices  $i$  and  $j$

If there are 4 matrices to be multiplied, how many ways can they be parenthesized?

☐ 4

☒ 5



☐ 14

☐ 6

In the Matrix Chain Multiplication problem, what is the significance of splitting the chain at position  $k$ ?

☒ It determines the optimal order of matrix multiplication



☐ It helps to find the matrix with the least number of rows

☐ It divides the chain into equal halves

☐ It ensures that no matrix is multiplied more than once

What is the parenthesization strategy in Matrix Chain Multiplication?

☐ To multiply matrices from left to right

☒ To explore all possible ways to parenthesize and choose the optimal one



☐ To divide the matrices into equal halves and multiply

☐ To multiply matrices from right to left



Question No: 15

**Multi Choice Type Question**

What is the primary objective of the Matrix Chain Multiplication problem?

- ☐ To multiply matrices in sequential order.
- ☐ To find the determinant of the matrices.
- ☐ To maximize the number of scalar multiplications.
- ☒ To minimize the number of scalar multiplications. ✓

How many recursive calls will be made when executing this code?

```
1 class MCM {
2     static int count = 0;
3     static int mcm(int[] arr, int i, int j) {
4         count++;
5         if (i == j) return 0;
6         int min = Integer.MAX_VALUE;
7         for (int k = i; k < j; k++) {
8             int cost = mcm(arr, i, k) + mcm(arr, k+1, j) + arr[i-1]*arr[k]*arr[j];
9             min = Math.min(min, cost);
10        }
11        return min;
12    }
13    public static void main(String[] args) {
14        int arr[] = {10, 20, 30, 40};
15        mcm(arr, 1, arr.length-1);
16        System.out.println(count);
17    }
18 }
```

☐ 8

☐ 5

☐ 10

☒ 9



What will be the minimal cost of multiplying these matrices?

```
1 public class MCM {
2     static int mcm(int[] arr, int i, int j) {
3         if(i == j) return 0;
4         int min = Integer.MAX_VALUE;
5         for(int k = i; k < j; k++) {
6             int cost = mcm(arr, i, k) + mcm(arr, k+1, j) + arr[i-1]*arr[k]*arr[j];
7             min = Math.min(min, cost);
8         }
9         return min;
10    }
11    public static void main(String[] args) {
12        int arr[] = {5, 10, 3, 12};
13        System.out.println(mcm(arr, 1, arr.length-1));
14    }
15 }
16
```

☒ 330

✓

☐ 270

What will be the output of the following code?

```
1 class MCM {
2     static int mcm(int p[], int i, int j) {
3         if (i == j) return 0;
4         int min = Integer.MAX_VALUE;
5         for (int k = i; k < j; k++) {
6             int cost = mcm(p, i, k) + mcm(p, k+1, j) + p[i-1]*p[k]*p[j];
7             if (cost < min) min = cost;
8         }
9         return min;
10    }
```



- ☐ 9000
- ☐ 5000
- ☐ 10000
- ☒ 6000 ✓

What will be the output of the following code?

```
1 class MCM {
2     static int mcm(int[] arr, int i, int j) {
3         if(i == j) return 0;
4         int min = Integer.MAX_VALUE;
5         for(int k = i; k < j; k++) {
6             int cost = mcm(arr, i, k) + mcm(arr, k+1, j) + arr[i-1]*arr[k]*arr[j];
7             min = Math.min(min, cost);
8         }
9         return min;
10    }
11    public static void main(String[] args) {
12        int arr[] = {5, 10, 3, 12};
13        System.out.println(mcm(arr, 1, arr.length-1));
14    }
15 }
```

☒ 156



☐ 189

☐ 144

☐ 200

What will be the output of the following code?

```
1 class MCM {
2     static int matrixChainOrder(int p[], int n) {
3         int m[][] = new int[n][n];
4         for (int L = 2; L < n; L++) {
5             for (int i = 1; i < n - L + 1; i++) {
6                 int j = i + L - 1;
7                 m[i][j] = Integer.MAX_VALUE;
8                 for (int k = i; k <= j - 1; k++) {
9                     int q = m[i][k] + m[k+1][j] + p[i-1]*p[k]*p[j];
10                    if (q < m[i][j]) m[i][j] = q;
11                }
12            }
13        }
14        return m[1][n-1];
15    }
16    public static void main(String args[]) {
17        int arr[] = {40, 20, 30, 10, 30};
18        System.out.println(matrixChainOrder(arr, arr.length));
19    }
20 }
21
```

☐ 60000

☐ 30000

☒ 26000



Which approach is generally preferred for solving the Matrix Chain Multiplication problem for large numbers of matrices?

- ☐ Brute-force approach
- ☒ Dynamic Programming ✓
- ☐ Divide and Conquer
- ☐ Greedy algorithm