

Multi Choice Type Question

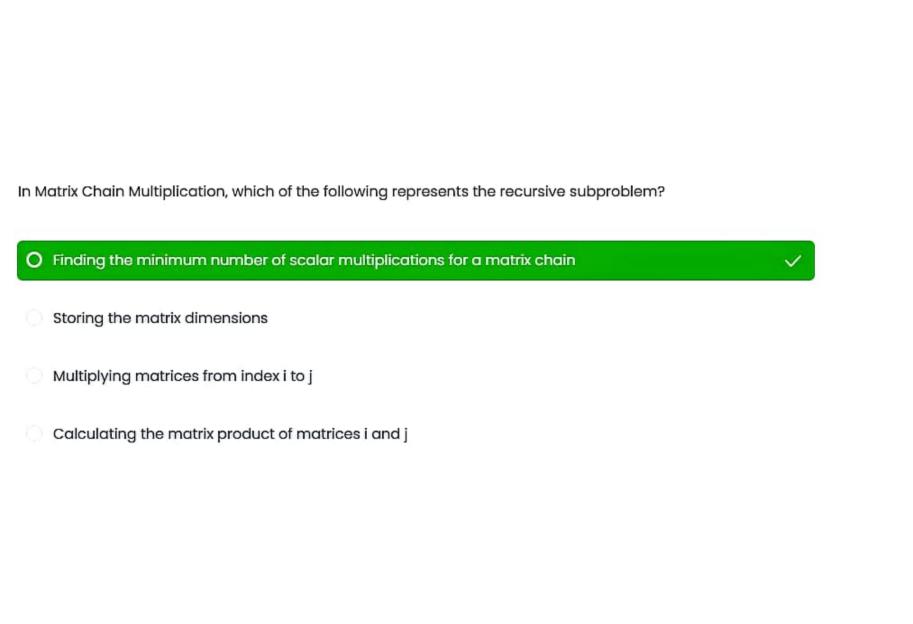
Blooms taxonomy Understand

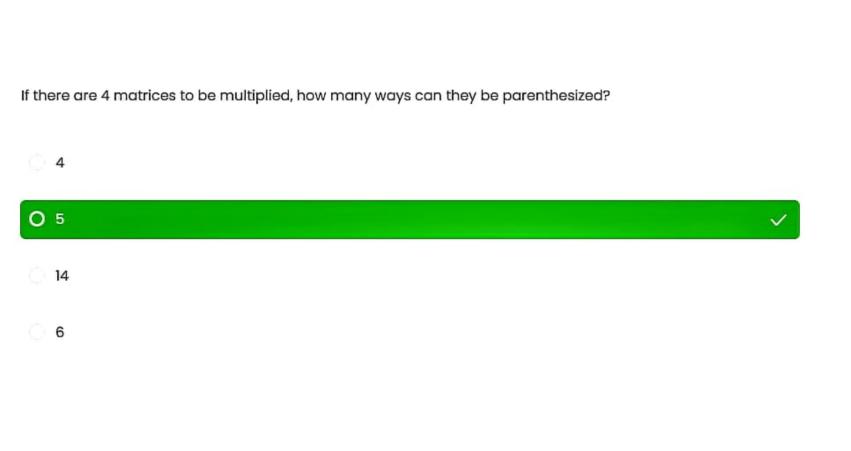
Show solution

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

2n
n-1
n

Status Correct Mark obtained 1/1 Hints used 0 Level Easy Question type MCQ Single Correct Subject Algorithms Topic Dynamic Programming Sub Topic Dynamic Programming





| n the Matrix Chain Multiplication problem, what is the significance of splitting the chain at position k? | | | | | |
|---|---|----------|--|--|--|
|) It de | etermines the optimal order of matrix multiplication | V | | | |
| It he | elps to find the matrix with the least number of rows | | | | |
| It di | vides the chain into equal halves | | | | |
| It er | sures that no matrix is multiplied more than once | | | | |
| | | | | | |
| | | | | | |
| | | | | | |

-1

| Who | at is the parenthesization strategy in Matrix Chain Multiplication? |
|-----|--|
| | To multiply matrices from left to right |
| 0 | To explore all possible ways to parenthesize and choose the optimal one |
| | |
| | To divide the matrices into equal halves and multiply |
| 0 | To divide the matrices into equal halves and multiply To multiply matrices from right to left |
| 0 | |
| 0 | |

| | Qu | estion | 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.
- O 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) {
            count++;
 4
 5
            if (i == j) return 0;
            int min = Integer.MAX_VALUE;
 6
            for (int k = i; k < j; k++) {
 7
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 }
```

10

8

What will be the minimal cost of multiplying these matrices?

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



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++) {
            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    }</pre>
```

made in the right in the same

curout println(mem are 1, are teneth this

- 9000
- 5000
- 10000

What will be the output of the following code?

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

O 156

189

144

What will be the output of the following code?

```
1 class MCM {
        static int matrixChainOrder(int p[], int n) {
 2
            int m[][] = new int[n][n];
 3
            for (int L = 2; L < n; L++) {
 4
                for (int i = 1; i < n - L + 1; i++) {
                    int j = i + L - 1;
 7
                    m[i][j] = Integer.MAX VALUE;
                    for (int k = i; k \le j - 1; k++) {
 8
 9
                        int q = m[i][k] + m[k+1][j] + p[i-1]*p[k]*p[j];
                        if (q < m[i][j]) m[i][j] = q;
10
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};
           System.out.println(matrixChainOrder(arr, arr.length));
18
19
        }
20 }
21
```

60000

