**Introduction to 2D Array**

[Report Issue](https://github.com/LeetCode-Feedback/LeetCode-Feedback/issues)

Similar to a one-dimensional array, a two-dimensional array also consists of a sequence of elements. But the elements can be laid out in a rectangular grid rather than a line.

*An Example*

*Java*

// "static void main" must be defined in a public class.

public class Main {

private static void printArray(int[][] a) {

for (int i = 0; i < a.length; ++i) {

System.out.println(a[i]);

}

for (int i = 0; i < a.length; ++i) {

for (int j = 0; a[i] != null && j < a[i].length; ++j) {

System.out.print(a[i][j] + " ");

}

System.out.println();

}

}

public static void main(String[] args) {

System.out.println("Example I:");

int[][] a = new int[2][5];

printArray(a);

System.out.println("Example II:");

int[][] b = new int[2][];

printArray(b);

System.out.println("Example III:");

b[0] = new int[3];

b[1] = new int[5];

printArray(b);

}

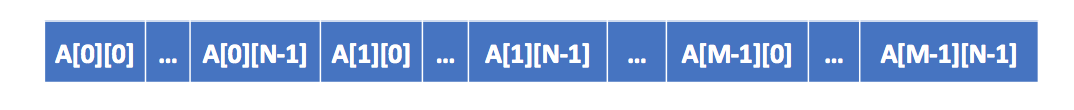
}

Let's take a look at an example of using a two-dimensional array:

*Principle*

In some languages, the multidimensional array is actually implemented internally as a one-dimensional array while in some other languages, there is actually no multidimensional array at all.

**1. C++ stores the two-dimensional array as a one-dimensional array.**

The picture below shows the actual structure of a *M \* N* array *A*:

So actually *A[i][j]* equals to *A[i \* N + j]* if we defined *A* as a one-dimensional array which also contains *M \* N* elements.

**2. In Java, the two-dimensional array is actually a one-dimensional array which contains *M* elements, each of which is an array of *N* integers.**

The picture below shows the actual structure of a two-dimensional array *A* in Java:

A diagram of a number

AI-generated content may be incorrect.

*Dynamic 2D Array*

Similar to the one-dimensional dynamic array, we can also define a dynamic two-dimensional array. Actually, it can be just a nested dynamic array. You can try it out by yourself.