

master ▾

...

leetcode-master / problems / 数组理论基础.md



youngyangyang04 更新头部信息

History

2 contributors



127 lines (79 sloc) | 4.8 KB

...

[PDF下载](#) [代码随想录](#) [刷题](#) [微信群](#) [B站](#) [代码随想录](#) [知识星球](#) [代码随想录](#)

欢迎大家[参与本项目](#)，贡献其他语言版本的代码，拥抱开源，让更多学习算法的小伙伴们收益！

数组理论基础

数组是非常基础的数据结构，在面试中，考察数组的题目一般在思维上都不难，**主要是考察对代码的掌控能力**

也就是说，想法很简单，但实现起来可能就不是那么回事了。

首先要知道数组在内存中的存储方式，这样才能真正理解数组相关的面试题

数组是存放在连续内存空间上的相同类型数据的集合。

数组可以方便的通过**下标索引**的方式获取到下标下对应的数据。

举一个字符数组的例子，如图所示：

内存地址：	100	101	102	103	104	105	106	107
字符数组：	S	A	B	J	H	J	A	B
下标：	0	1	2	3	4	5	6	7

需要两点注意的是

- 数组下标都是从0开始的。**

- 数组内存空间的地址是连续的

正是因为数组的在内存空间的地址是连续的，所以我们在删除或者增添元素的时候，就难免要移动其他元素的地址。

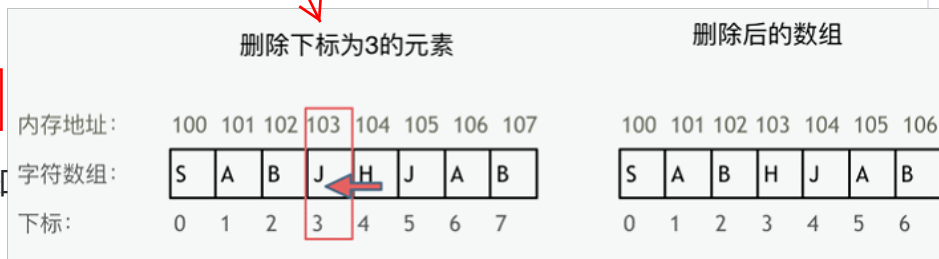
例如删除下标为3的元素，需要对下标为3的元素后面的所有元素都要做移动操作，如图所示：

算法通关数组1

而且大家如果使用C++的话，要注意vector 和 array的区别，vector的底层实现是array，严格来讲vector是容器，不是数组。

数组的元素是不能删的，只能覆盖。

那么二维数组直接上图，大家应该就知



		列 (第二索引)			
		0	1	2	3
行 (第一索引)	0	3	4	2	8
	1	4	5	6	2
	2	4	5	2	4

$a[0][1] = 1;$

$a[2][3] = 3;$

那么二维数组在内存的空间地址是连续的吗？

不同编程语言的内存管理是不一样的，以C++为例，在C++中二维数组是连续分布的。

我们来做实验，C++测试代码如下：

```
void test_arr() {
    int array[2][3] = {
        {0, 1, 2},
        {3, 4, 5}
    };
    cout << &array[0][0] << " " << &array[0][1] << " " << &array[0][2] << endl;
    cout << &array[1][0] << " " << &array[1][1] << " " << &array[1][2] << endl;
}

int main() {
    test_arr();
}
```

测试地址为

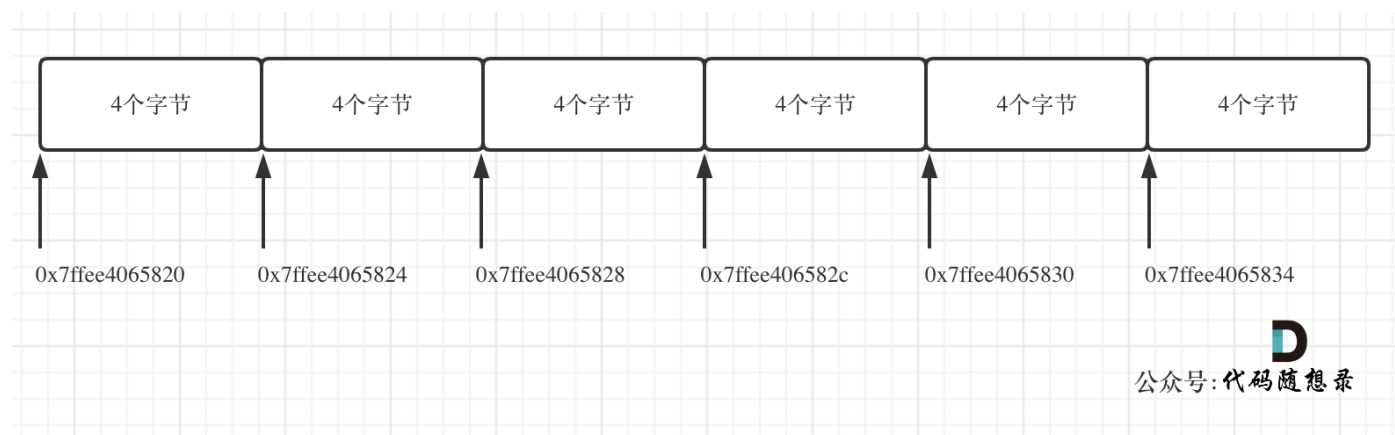
```
0x7ffee4065820 0x7ffee4065824 0x7ffee4065828
0x7ffee406582c 0x7ffee4065830 0x7ffee4065834
```

注意地址为16进制，可以看出二维数组地址是连续一条线的。

一些录友可能看不懂内存地址，我就简单介绍一下，0x7ffee4065820 与 0x7ffee4065824 差了一个4，就是4个字节，因为这是一个int型的数组，所以两个相信数组元素地址差4个字节。

0x7ffee4065828 与 0x7ffee406582c 也是差了4个字节，在16进制里8 + 4 = c，c就是12。

如图：



所以可以看出在C++中二维数组在地址空间上是连续的。

像Java是没有指针的，同时也不对程序员暴露其元素的地址，寻址操作完全交给虚拟机。

所以看不到每个元素的地址情况，这里我以Java为例，也做一个实验。

```
public static void test_arr() {  
    int[][] arr = {{1, 2, 3}, {3, 4, 5}, {6, 7, 8}, {9,9,9}};  
    System.out.println(arr[0]);  
    System.out.println(arr[1]);  
    System.out.println(arr[2]);  
    System.out.println(arr[3]);  
}
```

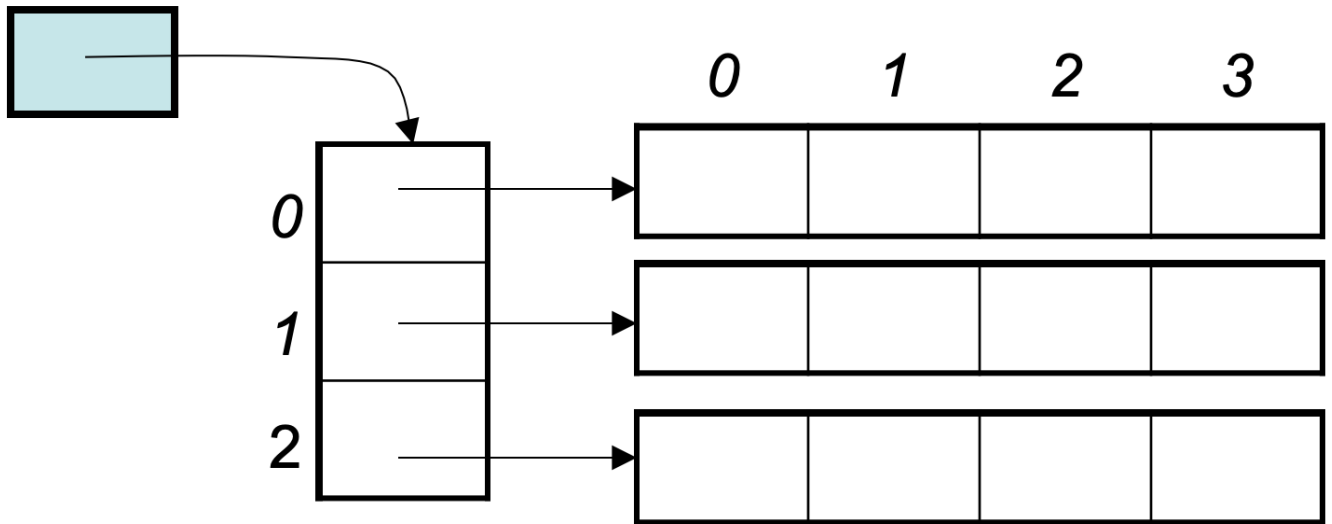
输出的地址为：

```
[I@7852e922  
[I@4e25154f  
[I@70dea4e  
[I@5c647e05
```

这里的数值也是16进制，这不是真正的地址，而是经过处理过后的数值了，我们也可以看出，二维数组的每一行头结点的地址是没有规则的，更谈不上连续。

所以Java的二维数组可能是如下排列的方式：

```
int[][] rating = new int[3][4];
```



这里面试中数组相关的理论知识就介绍完了。

- 作者微信: [程序员Carl](#)
- B站视频: [代码随想录](#)
- 知识星球: [代码随想录](#)