复习：

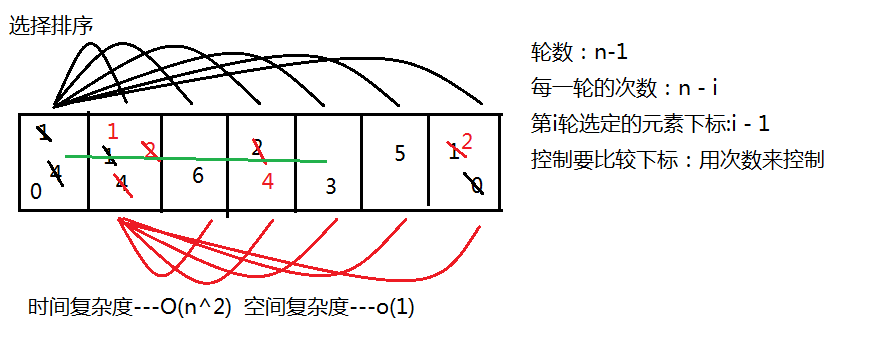
1. break和continue：break能用于选择或者是循环结构，表示终止当前一层结构，continue只能用于循环结构，表示跳出本次循环继续下一次

2. 数组：用于存储多个同一类型的数据的容器，大小固定。---下标。---数组的内存存储。

### 数组的应用

5. 数组排序

冒泡排序：相邻两个元素依次比较，如果是升序排序且前一个元素大于后一个元素，需要交换位置。---时间复杂度是O(n2), 空间复杂度是o(1)



时间复杂度: (n - 1)(n - i) = n2 - (i + 1)n + i -> O(n2)

空间复杂度：3 -> 3 \* n0 -> o(n0) -> o(1)

Arrays.sort(数组);--- 只能进行升序排序 --- 快速排序 --- O(nlogn)

6. 数组的反转 --- 头尾互换

7. 获取指定元素的下标---如果数组无序，通过遍历比较的方式来找到元素 --- 数组有序的话，可以使用二分查找 --- 时间复杂度O(logn), 空间复杂度o(1)

8. 数组的复制

System.arraycopy(要复制的数组，要复制的起始下标，要存放的数组，要存放的起始下标，要复制的元素个数);

原数组 = Arrays.copyOf(数组，长度); ---本质上就是数组的复制---必然会产生一个新的数组，所以数组扩容完成之后地址一定会发生改变

## 二维数组

存储数组的数组---里面所存储的元素都是一维数组。

### 定义格式

数据类型[][] 数组名 = new 数据类型[二维数组的大小][一维数组的大小];

int[][] arr = new int[3][5]; --- 表示定义了一个能存储3个一维数组的二维数组，每一个一维数组中能存储5个整型元素

int[] is = arr[1];

arr[1][3] = 5;

int i = arr[1][3];

数据类型[][] 数组名 = new 数据类型[二维数组的大小][];

int[][] arr = new int[5][]; --- 表示这个二维数组中能存储5个一维数组

arr[0] = new int[3];

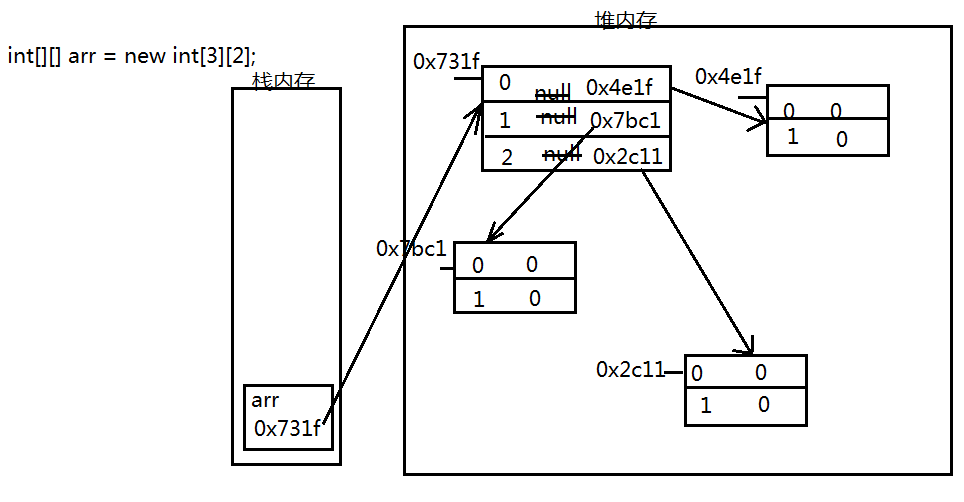
arr[1] = new int[7];

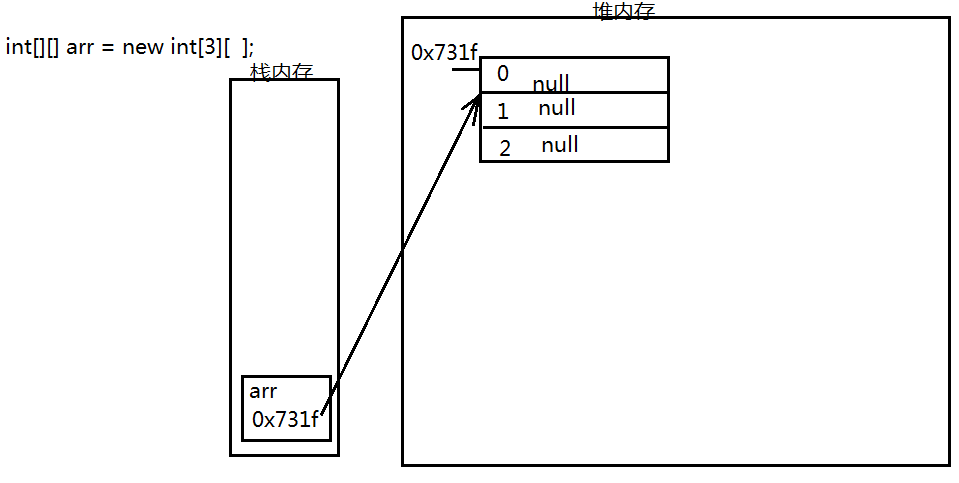
arr[2] = new int[2];

数据类型[][] 数组名 = {{数组1}，{数组2}， {数组3} ...};

int[][] arr = {{3,1} , {6,8,4,2,7}, {1}, {5,9,5,3}}; --- 表示定义了整型的二维数组，大小为4，而且每一个一维数组中的元素是已知的

### 二维数组的内存存储





思考：有int[] x[] , y;均已初始化，下列各项中正确的是：-- BE

A. x = y;

B. x[0] = y ;

C. x[0][0] = y;

D. x[0] = y[0];

E. x[0][0] = y[0];

思考：

1. 一维数组中无序存放了1~n中的n-1个数，如何找到这一个没有放到数组中的数？--- O(n) o(1) --- +

2. 一个数组存放了一组数字，除了其中的一个元素以外，其余的元素都出现了偶数次，只有一个元素出现了奇数次，找到这个出现了奇数次的元素 --- O(n) o(1) --- ^

1,1,2,2,2,3,3,3,3,3,2,2,2,4,4,4,4,4,4,5,5,5,5,6,6,7,7,7,8,7,8,9,8,9,8...

int n = 0;

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

n ^= arr[i];

}

练习：

1. 猴子选大王： 有15只猴子围成了一圈选大王，每只猴子依次报数，报到7的猴子就被淘汰，剩余1只猴子，那么这只猴子就是大王。---这只成为大王的猴子是第几只？

2. 杨辉三角：

1

1 1

1 2 1

1 3 3 1

1 4 6 4 1

1 5 10 10 5 1

1 6 15 20 15 6 1

arr[i][j] = arr[i - 1][j] + arr[i - 1][j - 1];

输入一个数字，输出对应的行数

## 方法

将要重复使用的逻辑或者是代码提取出来写成一个新的形式--- 方法

修饰符 返回值类型 方法名(参数列表){

方法体;

return 返回值;

}

练习：

1. 哥德巴赫猜想：任何一个大于等于6的偶数都可以分解成两个质数之和

6 = 3 + 3

18 = 5 + 13

18 = 7 + 11

输入一个数字，输出这个数字的所有的分解形式

14 = 3 + 11

14 = 7 + 7

2. 亲密数：如果A的所有因子（不含本身）之和等于B，而且B的所有因子（不含本身）之和等于A，那么A和B就是一对亲密数---输出5000以内所有的亲密数

8 ： 1 + 2 + 4 = 7

7 ： 1 = 1

因为1 != 8, 所以7和8不是亲密数

方法如果没有返回值，就将返回值类型定义为void

方法的返回值类型是void，是否意味着这个方法中不能写return语句？ -- 可以 --- return;

方法的定义过程需要明确返回值类型，明确参数