1.一个按升序排列好的数组int[] arry = {-5,-1,0,5,9,11,13,15,22,35,46},输入一个x，int x = 31，在数据中找出和为x的两个数，例如 9 + 22 = 31，要求算法的时间复杂度为O(n);
2.如何向一个数据库具有int类型A，B，C，D四列的表中随机插入10000条数据？如何按升序取出A列中前10个数?
3.x个苹果，一天只能吃一个、两个、或者三个，问多少天可以吃完
4.一个无序，不重复数组，输出N个元素，使得N个元素的和相加为M，给出时间复杂度、空间复杂度。手写算法

1.一个按升序排列好的数组int[] arry = {-5,-1,0,5,9,11,13,15,22,35,46},输入一个x，int x = 31，在数据中找出和为x的两个数，例如 9 + 22 = 31，要求算法的时间复杂度为O(n);

分析：该题不难，主要关注点应该为要求时间复杂度为O(n),因为数组是按升序排列，所以可以定义两个指针i、j，分别从数组的两端开始遍历，如果a[i]+a[j]大于31，则应该让尾指针j前移，如果a[i]+a[j]小于31，则应该让头指针i后移，直到找到a[i]+a[j]等于31，或遍历完成

```
public class Find {
    public static void main(String[] args) {
        int[] arr = {-5, -1, 0, 5, 9, 11, 13, 15, 22, 35, 46};
        int sum = 31;
        find(arr, sum);
    }
    private static void find(int[] arr, int sum) {
        if (arr.length <= 1) {
            System.out.println("arr wrong");
            return;
        }
        int i = 0;
        int j = arr.length - 1;
        while (i != j) {
            n++;
            int tmpSum = arr[i] + arr[j];
            if (tmpSum == sum) {
                System.out.println("a[" + i + "] = " + arr[i] + ", a[" + j + "] = " + arr[j]);
                return;
            }
            if (tmpSum < sum) i++;
            if (tmpSum > sum) j--;
        }
        System.out.println("not found");
```

```
    }
}

```

2.如何向一个数据库具有int类型A，B，C，D四列的表中随机插入10000条数据？如何按升序取出A列中前10个数？

说明: 1、随机数可以在代码生成，开启事务之后循环插入，然后关闭事务。

2、使用limit和order by进行升序取固定个数的值

3.x个苹果，一天只能吃一个、两个、或者三个，问多少天可以吃完

4.一个无序，不重复数组，输出N个元素，使得N个元素的和相加为M，给出时间复杂度、空间复杂度。手写算法

排序方法	时间复杂度（平均）	时间复杂度（最坏）	时间复杂度（最好）	空间复杂度	稳定性
插入排序	$O(n^2)$	$O(n^2)$	$O(n)$	$O(1)$	稳定
希尔排序	$O(n^{1.3})$	$O(n^2)$	$O(n)$	$O(1)$	不稳定
选择排序	$O(n^2)$	$O(n^2)$	$O(n^2)$	$O(1)$	不稳定
堆排序	$O(nlog_2n)$	$O(nlog_2n)$	$O(nlog_2n)$	$O(1)$	不稳定
冒泡排序	$O(n^2)$	$O(n^2)$	$O(n)$	$O(1)$	稳定
快速排序	$O(nlog_2n)$	$O(n^2)$	$O(nlog_2n)$	$O(nlog_2n)$	不稳定
归并排序	$O(nlog_2n)$	$O(nlog_2n)$	$O(nlog_2n)$	$O(n)$	稳定
计数排序	$O(n+k)$	$O(n+k)$	$O(n+k)$	$O(n+k)$	稳定
桶排序	$O(n+k)$	$O(n^2)$	$O(n)$	$O(n+k)$	稳定
基数排序	$O(n*k)$	$O(n*k)$	$O(n*k)$	$O(n+k)$	稳定