1. 无重复的最长重复字串

优化思路：在暴力法中我们要检查一个字符串中是否包含有重复的字符，需要遍历查询，复杂度太高；我们本次可以使用hashset来查询时间复杂度为O(1)。

本次复杂度为O（2n）到 O（n）

public int lengthOfLongestSubstring(String s) {

int n = s.length();

Set<Character> set = new HashSet<>();

int ans = 0, i = 0, j = 0;

while (i < n && j < n) {

// try to extend the range [i, j]

if (!set.contains(s.charAt(j))){

set.add(s.charAt(j++));

ans = Math.max(ans, j - i);

}

else {

set.remove(s.charAt(i++));

}

}

return ans;

}

1. 寻找两个有序数组的中位数

要求时间复杂度为O（log（m+n））

由于这个数组是有序的，所以我们首先想到的就是使用分治的思想。把一个数组分为两半，保证最左边最大的数小于最右边最小的数。

然后我们将 A的前半部分和B的前半部分 合在一起，把A的后半部分和B的后半部分合在一起， 假如点半部分合起来的最大值 小于等于后半部分的最小值，那么中位数不是直接就求出了了么。用一个关系式表示：

j+i=(n+m)/2 && max(A[j-1] B[i-1]) <= min(A[j] B[i])

1. 买卖股票的最佳时机（最多可以交易两次）

思想：动态规划 + 二分法

以i天为分界线；计算i天之前的最大交易和i天之后的最大交易值。

然后求最大值max（left + right）