题目

Given an array of integers, return **indices** of the two numbers such that they add up to a specific target.

You may assume that each input would have ***exactly*** one solution, and you may not use the *same* element twice.

**Example:**

Given nums = [2, 7, 11, 15], target = 9,

Because nums[**0**] + nums[**1**] = 2 + 7 = 9,

return [**0**, **1**].

给你一个目标，它将是ARRAY中两个数字加起来，让你return这两个数字的Index

class Solution {

public int[] twoSum(int[] nums, int target) {

HashMap<Integer,Integer> hashMap = new HashMap<>();

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

int diff = target - nums[i];

if (hashMap.containsKey(diff)){

return new int[] {hashMap.get(diff),i};

}

hashMap.put(nums[i], i);

}

return nums;

}

}

思路：用两个for就行，但是这样时间复杂度是n^2,

最好的解使用hashMap,.

先算diff，如果这个hashMap存在diff， return

如果不存在，把 值,i 加入Hashmap

那么loop到最后所有数字都会存进去，到最后从能匹配到

学到的点：

hashMap 加入, put(key,value);

提取get(key)

存在：containsKey(). containsValue

HashMap<int,int>不行，必须是Integer