# 题目

给你一个下标从0开始的整数数组nums。请你从nums中找出和最大的一对数，且这两个数数位上最大的数字相等。

返回最大和，如果不存在满足题意的数字对，返回-1。

示例 1：

输入：nums = [51,71,17,24,42]

输出：88

解释：

i = 1和j = 2，nums[i]和nums[j]数位上最大的数字相等，且这一对的总和71 + 17 = 88。

i = 3和j = 4，nums[i]和nums[j]数位上最大的数字相等，且这一对的总和24 + 42 = 66。

可以证明不存在其他数对满足数位上最大的数字相等，所以答案是88。

示例 2：

输入：nums = [1,2,3,4]

输出：-1

解释：不存在数对满足数位上最大的数字相等。

提示：

2 <= nums.length <= 100

1 <= nums[i] <= 10^4

# 分析

## 方法一：哈希表

思路：

使用一个长度为10的vector来记录每个数位上最大的数字对应的数值。具体解题思路如下：

1、创建一个长度为10的vector，初始化为-1，用来记录每个数位上最大的数字对应的数值。

2、遍历输入的数组nums，对于数组中的每个数字，找出其数位上最大的数字。

3、如果当前数位上最大的数字在vector中对应的位置不为-1（即已经存在一个数和当前数的数位上最大的数字相等），则更新结果res为当前数和vector中对应位置的数的和的较大值。

4、将当前数和vector中对应位置的数的较大值更新到vector中对应位置。

5、返回结果res作为最终答案。

这样就可以找出和最大的一对数，且这两个数数位上最大的数字相等。

代码：

class Solution {

public:

int maxSum(vector<int>& nums) {

vector<int> vec(10, -1);

int res = -1;

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

int n = nums[i], max = -1;

// 用来计算一个整数的各个数位上的最大数字的

while (n) {

max = std::max(max, n % 10);

n /= 10;

}

if (vec[max] != -1) {

res = std::max(res, vec[max] + nums[i]);

}

vec[max] = std::max(vec[max], nums[i]);

}

return res;

}

};