# 题目

给你一个整数数组 digits ，其中每个元素是一个数字（0 - 9）。数组中可能存在重复元素。

你需要找出 所有 满足下述条件且 互不相同 的整数：

该整数由 digits 中的三个元素按 任意 顺序 依次连接组成。

该整数不含前导零

该整数是一个偶数

例如，给定的 digits 是 [1, 2, 3] ，整数 132 和 312 满足上面列出的全部条件。

将找出的所有互不相同的整数按 递增顺序 排列，并以数组形式返回。

示例 1：

输入：digits = [2,1,3,0]

输出：[102,120,130,132,210,230,302,310,312,320]

解释：

所有满足题目条件的整数都在输出数组中列出。

注意，答案数组中不含有 奇数 或带 前导零 的整数。

示例 2：

输入：digits = [2,2,8,8,2]

输出：[222,228,282,288,822,828,882]

解释：

同样的数字（0 - 9）在构造整数时可以重复多次，重复次数最多与其在 digits 中出现的次数一样。

在这个例子中，数字 8 在构造 288、828 和 882 时都重复了两次。

示例 3：

输入：digits = [3,7,5]

输出：[]

解释：

使用给定的 digits 无法构造偶数。

提示：

3 <= digits.length <= 100

0 <= digits[i] <= 9

# 分析

## 方法一：暴力法

思路：

代码：

class Solution {

public:

vector<int> findEvenNumbers(vector<int>& digits) {

unordered\_set<int> nums; // 目标偶数集合

int n = digits.size();

// 遍历三个数位的下标

for (int i = 0; i < n; ++i){

for (int j = 0; j < n; ++j){

for (int k = 0; k < n; ++k){

// 判断是否满足目标偶数的条件

if (i == j || j == k || i == k){

continue;

}

int num = digits[i] \* 100 + digits[j] \* 10 + digits[k];

if (num >= 100 && num % 2 == 0){

nums.insert(num);

}

}

}

}

// 转化为升序排序的数组

vector<int> res;

for (const int num: nums){

res.push\_back(num);

}

sort(res.begin(), res.end());

return res;

}

};