# 题目

给定一个二进制数组nums (索引从0开始)。

我们将xi定义为其二进制表示形式为子数组nums[0..i] (从最高有效位到最低有效位)。

例如，如果nums =[1,0,1] ，那么 x0 = 1, x1 = 2, 和 x2 = 5。

返回布尔值列表answer，只有当xi可以被5整除时，答案answer[i]为true，否则为false。

示例 1：

输入：nums = [0,1,1]

输出：[true,false,false]

解释：

输入数字为0, 01, 011；也就是十进制中的0, 1, 3 。只有第一个数可以被5整除，因此answer[0]为true。

示例 2：

输入：nums = [1,1,1]

输出：[false,false,false]

提示：

1 <= nums.length <= 105

nums[i]仅为0或1

# 分析

思路：

代码：

class Solution {

public:

vector<bool> prefixesDivBy5(vector<int>& nums) {

vector<bool> ans;

int length = nums.size();

int prefix = 0;

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

// 每次都要左移1位，然后加上个位数，构造为新的数

prefix = ((prefix << 1) + nums[i]) % 5;

if (prefix == 0) {

ans.push\_back(true);

} else {

ans.push\_back(false);

}

}

return ans;

}

};

复杂度分析：

时间复杂度：O(n)，其中n是数组nums的长度。需要遍历数组一次并计算前缀。

空间复杂度：O(1)。除了返回值以外，额外使用的空间为常数。