# 题目

给你一个正整数的数组 A。

然后计算 S，使其等于数组 A 当中最小的那个元素各个数位上数字之和。

最后，假如 S 所得计算结果是 奇数 ，返回 0 ；否则请返回 1。

示例 1:

输入：[34,23,1,24,75,33,54,8]

输出：0

解释：

最小元素为 1 ，该元素各个数位上的数字之和 S = 1 ，是奇数所以答案为 0 。

示例 2：

输入：[99,77,33,66,55]

输出：1

解释：

最小元素为 33 ，该元素各个数位上的数字之和 S = 3 + 3 = 6 ，是偶数所以答案为 1 。

提示：

1 <= A.length <= 100

1 <= A[i] <= 100

# 分析

class Solution {

public:

int sumOfDigits(vector<int>& A) {

int min\_val = A.at(0);

for (int i = 1; i < A.size(); ++i)

{

min\_val = min(min\_val, A.at(i));

}

int s = 0;

while (min\_val)

{

s += min\_val % 10;

min\_val /= 10;

}

return s % 2 == 0;

}

};

排序获得最小元素，然后将其每一位相加即可，判断奇偶。

class Solution {

public:

int sumOfDigits(vector<int>& A) {

int min\_val = 100;

for (int x : A) min\_val = min(min\_val, x);

int res = 0;

while (min\_val)

{

res += min\_val % 10;

min\_val /= 10;

}

if (res % 2) return 0;

return 1;

}

};

class Solution

{

public:

int sumOfDigits(vector<int>& A)

{

int x = \*min\_element(A.begin(), A.end());

int res = 0;

while(x)

{

res += (x % 10);

x /= 10;

}

return (res % 2 == 1 ? 0 : 1);

}

};

class Solution

{

public:

int sumOfDigits(vector<int>& A)

{

int x = \*min\_element(A.begin(), A.end());

int res = 0;

while(x)

{

res += (x % 10);

x /= 10;

}

return (res % 2 == 1 ? 0 : 1);

}

};

inline int dsum(int x){

int s=0;

while(x){s+=x%10;x/=10;}

return s;

}

class Solution {

public:

int sumOfDigits(vector<int>& a) {

int n=a.size();

int mi=a[0];

for(int i=1;i<n;++i)mi=min(mi,a[i]);

return dsum(mi)&1^1;

}

};

复杂度分析

时间复杂度：O(n)

空间复杂度：O(1)