

202. Happy Number

☰ Tags	
📅 Property	
🔗 URL	

Question

原文：

Write an algorithm to determine if a number `n` is happy.

A **happy number** is a number defined by the following process:

- Starting with any positive integer, replace the number by the sum of the squares of its digits.
- Repeat the process until the number equals 1 (where it will stay), or it **loops endlessly in a cycle** which does not include 1.
- Those numbers for which this process **ends in 1** are happy.

Return `true` if `n` is a happy number, and `false` if not.

我的理解：

假設進來的數字是19，要拆成 $1^2+9^2=82$ ，如果不等於 1 繼續拆並且都是平方相加， $8^2+2^2=68$ ， $6^2+8^2=100$ ， $1^2+0^2+0^2=1$ ，如果最終能像19等於 1 就是 happy number return true，如果會無止境的算下去就是 false

翻譯：

写一个算法来确定一个数字n是否是快乐的。

一个快乐的数字是一个由以下过程定义的数字。

从任何一个正整数开始，用其数字的平方之和替换该数字。

重复这个过程，直到这个数字等于1（它将停留在这个位置），或者它在一个不包括1的循环中无休止地循环。

这个过程以1结束的那些数字是快乐的。

如果n是一个快乐的数字，返回true；如果不是，返回false。

自評翻譯正確性：90

- Word Memory :
 - endlessly 無止境地

Code

```
class Solution {
public:
    bool isHappy(int n) {
        int i=0,temp=0,nsquare=0;
        vector<int> store;
        while(nsquare!=1){
            nsquare=0;
            while(n>0){
                nsquare=nsquare+(n%10)*(n%10);//計算當前n的平方總和
                n=n/10;
            }
            n=nsquare;//nsquare就是下一個n
            for(i=0;i<store.size();i++){//如果nsquare是之前出現過的數字表示計算已經陷入無盡迴圈
                if(nsquare==store[i]){
                    return false;
                }
            }
            store.push_back(nsquare);//將新的nsquare放進store
        }
        return true;
    }
};
```

思路：先把各位數拆分做平方和相加nsquare，然後紀錄nsquare的數值，拿去跟store中的數值比較，如果是之前出現過的數值表示計算已經陷入無盡迴圈，那此時就直接false，如果等到nsquare等於1那就出迴圈 return true

Success Details >

Runtime: 3 ms, faster than 64.76% of C++ online submissions for Happy Number.

Memory Usage: 6.2 MB, less than 42.30% of C++ online submissions for Happy Number.

Next challenges:

Add Digits

Ugly Number

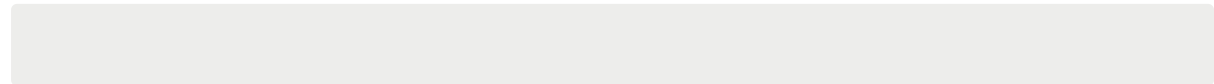
Sum of Digits of String After Convert

Show off your acceptance:



Time Submitted	Status	Runtime	Memory	Language
10/06/2022 09:57	Accepted	3 ms	6.2 MB	cpp

優良code參考



思路：