

# 645. Set Mismatch範例

Tags	unordered_map
Property	@October 23, 2022
URL	

## Question

原文：

You have a set of integers `s`, which originally contains all the numbers from `1` to `n`. Unfortunately, due to some error, one of the numbers in `s` got duplicated to another number in the set, which results in **repetition of one** number and **loss of another** number.

You are given an integer array `nums` representing the data status of this set after the error.

Find the number that occurs twice and the number that is missing and return *them in the form of an array*.

我的理解：

給定一個陣列，陣列中理論上有1~n個數字，且包含1~n，但不幸的有一個數字重複，也意味有一個數字丟失不再陣列中

return 重複的數字跟丟失的數字

翻譯：

你有一个整数集合s，它最初包含了从1到n的所有数字。不幸的是，由于一些错误，s中的一个数字被重复到了集合中的另一个数字上，这导致了一个数字的重复和另一个数字的丢失。

我们给你一个整数数组nums，代表错误发生后这个集合的数据状态。

找到出现两次的数字和丢失的数字，并以数组的形式返回。

自評翻譯正確性：70

- Word Memory：

# Code

```
class Solution {
public:
    vector<int> findErrorNums(vector<int>& nums) {
        int i = 0;
        unordered_map<int,int>temp;
        vector<int>ans;
        for(i=0;i<nums.size();i++){
            if(temp.count(nums[i])==0){
                temp[nums[i]]=1;
            }
            else if(temp.count(nums[i])==1){
                temp[nums[i]]++;
                ans.push_back(nums[i]);
            }
        }
        for(i=0;i<nums.size();i++){
            if(temp.count(i+1)==0){
                ans.push_back(i+1);
            }
        }
        return ans;
    }
};
```

思路：

- 利用hashmap把出現過的數字轉成字典，如果已經出現過又再次出現，表示該**數字重複**，先push進vector
- 第二個for，從1~n開始檢查，如果有數字沒有被建立字典，表示該**數字在字典沒有出現過**，push進vector

**優良code參考 (另一種思路，速度較快，但好像是快在不用hash map)**

```
class Solution {
public:
    vector<int> findErrorNums(vector<int>& nums) {
        int i = 0, x=0;
        int sum = ((1+nums.size())*nums.size())/2;
        //unordered_map<int,int>temp;
        int temp[10001]={0};
        vector<int>ans;
        for( i=0 ; i<nums.size() ; i++){
            if(temp[nums[i]]==0){
```

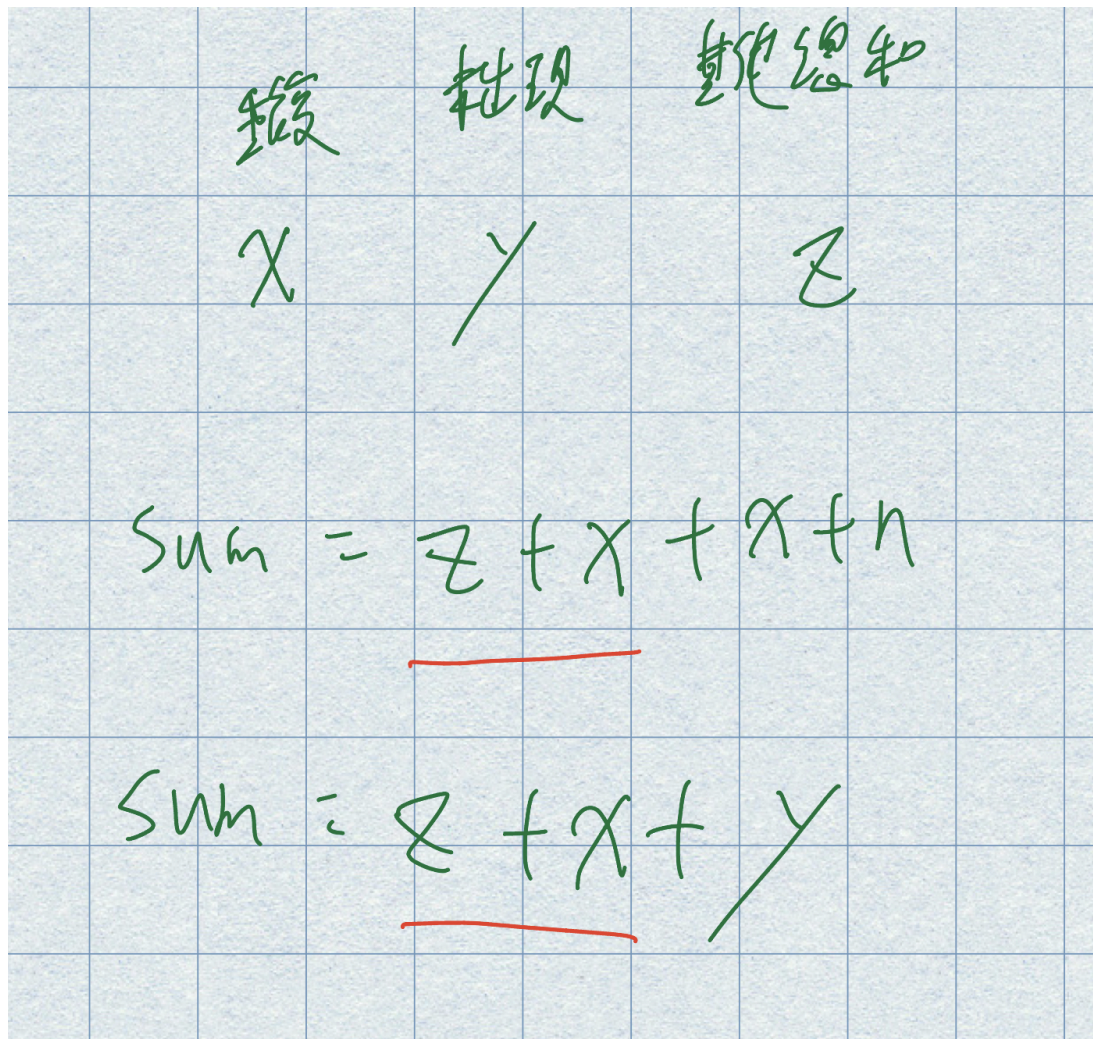
```

        sum-=nums[i];
        temp[nums[i]]++;
    }
    else if(temp[nums[i]]==1){
        x=nums[i];
        ans.push_back(nums[i]);
    }
}
ans.push_back(sum);
return ans;
}
};

```

思路：

- (也是我寫的但參考了別人的想法)
- 因為陣列是 1~n 各有一個，陣列中如果沒有重複的數字，總和可以透過梯形公式計算
  - $(上底 + 下底 * 高) / 2$
- 下方for，開始從 sum 中把每個數字減去，如果出現重複數字則不計算，並且加入vector
- 最後減去陣列中所有的值（不重複減去重複的數字），剩下來的sum值就是那個被遺漏的數字



Success Details >

Runtime: 61 ms, faster than 71.01% of C++ online submissions for Set Mismatch.

Memory Usage: 21.5 MB, less than 50.70% of C++ online submissions for Set Mismatch.

Next challenges:

Find the Duplicate Number

Show off your acceptance:



Time Submitted	Status	Runtime	Memory	Language
10/23/2022 17:14	Accepted	61 ms	21.5 MB	cpp