

# 869. Reordered Power of 2

☰ Tags	medium
📅 Property	@August 29, 2022

## Question

原文：

You are given an integer `n`. We reorder the digits in any order (including the original order) such that the leading digit is not zero.

Return `true` if and only if we can do this so that the resulting number is a power of two.

我的理解：

給定一個數字n，並會將n的數字重新排序例如n等於121，可能被重新排序成211、121、112，但如果n中有0的畫0不會因為重新排序就被放到第一位（換言之位數不會降低，10不會重新排序成01）

如果這些重新排序後的結果跟任何一個2的次方相等則return true，否則false

翻譯：

我们以任何顺序（包括原来的顺序）重新排列数字，使前导数字不为零。

当且仅当我们能做到这一点，使结果的数字是2的幂时，返回true。

自評翻譯正確性：100%

- Word Memory：

## Code

```
class Solution {
public:
    bool reorderedPowerOf2(int n) {
        int twopow[31]={0};
        string twopowstr[31];
        int i;
        for(i=0;i<31;i++){//計算2的次方
```

```

        twopow[i]=pow(2,i);
    }
    for(i=0;i<31;i++){
        twopowstr[i]=to_string(twopow[i]); //int 轉 string
        sort(twopowstr[i].begin(),twopowstr[i].end()); //string再做排序
    }
    string strn=to_string(n);
    sort(strn.begin(),strn.end());
    for(i=0;i<31;i++){
        if(twopowstr[i]==strn){ //比對兩個以排序後構造是否相同
            return true;
        }
    }
    return false;
}
};

```

思路：先算出n可能的大小區間（ $1 \sim 10^9$ ）內所有2個次方，並且將其轉成字串並排序，這部是為了直接看出構造，如果n也轉成字串並排序後構造相同，則表示n重新排序後會有一個組合等同於那個構造相同的2的次方

Success [Details >](#)

Runtime: 0 ms, faster than 100.00% of C++ online submissions for Reordered Power of 2.

Memory Usage: 6.1 MB, less than 63.91% of C++ online submissions for Reordered Power of 2.

Next challenges:

Smallest Good Base

Smallest Range I

Minimum Number of Lines to Cover Points

Show off your acceptance:



Time Submitted	Status	Runtime	Memory	Language
08/29/2022 11:43	Accepted	0 ms	6.1 MB	cpp

## 優良code參考

```
class Solution {
public:
    bool reorderedPowerOf2(int n) {
        string given = to_string(n);
        sort(given.begin(), given.end());

        for(int i = 0; i < 30; i++){
            string temp = to_string(1 << i);
            sort(temp.begin(), temp.end());
            if(given == temp){
                return true;
            }
        }
        return false;
    }
};
```

思路：大致相同，但參考了他sort的用法