401 Binary Watch

2018年4月10日 18:55

Question:

A binary watch has 4 LEDs on the top which represent the **hours** (**0-11**), and the 6 LEDs on the bottom represent the **minutes** (**0-59**).
Each LED represents a zero or one, with the least significant bit on the right.

For example, the above binary watch reads "3:25".

Given a non-negative integer *n* which represents the number of LEDs that are currently on, return all possible times the watch could represent.

Example:

Input: n = 1

Return: ["1:00", "2:00", "4:00", "8:00", "0:01", "0:02", "0:04", "0:08", "0:16", "0:32"]

Note:

- The order of output does not matter.
- The hour must not contain a leading zero, for example "01:00" is not valid, it should be "1:00".
- The minute must be consist of two digits and may contain a leading zero, for example "10:2" is not valid, it should be "10:02".

来自 <https://leetcode.com/problems/binary-watch/description/>

二进制手表顶部有 4 个 LED 代表**小时（0-11）**，底部的 6 个 LED 代表**分钟（0-59）**。

每个 LED 代表一个 0 或 1，最低位在右侧。

例如，上面的二进制手表读取 “3:25”。

给定一个非负整数 *n* 代表当前 LED 亮着的数量，返回所有可能的时间。

Solution for Python3:

```python
class Solution:
    def readBinaryWatch(self, num):
        """
        :type num: int
        :rtype: List[str]
        """
        return ['%d:%02d' % (h, m) for h in range(12) for m in range(60) if (bin(h) + bin(m)).count('1') == num]
```

```cpp
class Solution {
public:
    vector<string> readBinaryWatch(int num) {
        vector<string> res;
        for (int h = 0; h < 12; h++) {
            for (int m = 0; m < 60; m++) {
                if (bitset<10>(h << 6 | m).count() == num) {
                    res.emplace_back(to_string(h) + (m < 10 ? ":0" : ":") + to_string(m));
                }
            }
        }
```

```
13          return res;
14      }
    };
```