

```

1  /*
2  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）。
3
4  8 4 2 1,32 16 8 4 2 1
5
6  每个 LED 代表一个 0 或 1，最低位在右侧。
7
8  例如，上面的二进制手表读取 “3:25”。
9
10 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。
11
12 案例：
13
14 输入：n = 1
15 返回：["1:00", "2:00", "4:00", "8:00", "0:01", "0:02", "0:04", "0:08",
16        "0:16", "0:32"]
17
18
19 注意事项：
20
21 输出的顺序没有要求。
22 小时不会以零开头，比如 “01:00” 是不允许的，应为 “1:00”。
23 分钟必须由两位数组成，可能会以零开头，比如 “10:2” 是无效的，应为 “10:02”。
24
25 来源：力扣（LeetCode）
26 链接：https://leetcode-cn.com/problems/binary-watch
27 著作权归领扣网络所有。商业转载请联系官方授权，非商业转载请注明出处。
28 */

```

分析:

- 方法一:回溯法,注意返回和边界条件控制.
- 方法二:二进制位法,利用10bit使用位操作来模拟取值.

方法一:回溯法

```

1  class Solution
2  {
3
4      private:
5
6          void helper(    vector<string>&    vs            ,
7                        int                num            ,
8                        int                cur_num         ,
9                        int                cur_hour        ,
10                       int                cur_minute     ,
11                       int                cur_index       ,
12                       int                (&arr)[10]
13                       )
14      {
15

```

```

16         if(num < cur_num || cur_hour > 11 || cur_minute > 59)
17         {
18             return;
19         }
20
21         if(num == cur_num)
22         {
23             char buf_hour[3]    = {0,};
24             char buf_minute[3]  = {0,};
25             sprintf(buf_hour, "%d", cur_hour);
26             sprintf(buf_minute, "%02d", cur_minute);
27             vs.push_back( buf_hour + string(":") + buf_minute);
28             return;
29         }
30
31         if(cur_index >= 10)
32         {
33             return;
34         }
35         // 第idx灯不选中
36         helper(vs,num,cur_num,cur_hour,cur_minute,cur_index+1,arr);
37         // 第idx灯选中
38         if(cur_index<4)
39         {
40
41             helper(vs,num,cur_num+1,cur_hour+arr[cur_index],cur_minute,cur_index+1,arr
42             );
43             }
44             else
45             {
46
47                 helper(vs,num,cur_num+1,cur_hour,cur_minute+arr[cur_index],cur_index+1,arr
48                 );
49             }
50         }
51
52     public:
53     vector<string> readBinarywatch(int num)
54     {
55
56         vector<string> vs;
57         int cur_hour;
58         int cur_minute;
59         int arr[10] = {8,4,2,1,32,16,8,4,2,1};
60
61         helper(vs,num,0,0,0,0,arr);
62         return vs;
63     }
64 };
65
66 /*
67 执行结果:
68 通过
69 显示详情

```

```
70 执行用时 :0 ms, 在所有 cpp 提交中击败了100.00% 的用户
71 内存消耗 :8.8 MB, 在所有 cpp 提交中击败了47.37%的用户
72 */
```

方法二:二进制位法

```
1  class solution
2  {
3
4      private:
5          int bitcount(unsigned short int n)
6          {
7              int count = 0;
8              while(n != 0){
9                  if(n%2 ==1)
10                     {
11                         count++;
12                     }
13                     n /= 2;
14             }
15             return count;
16         }
17
18
19     public:
20         vector<string> readBinarywatch(int num)
21         {
22
23             vector<string> vs ;
24             int cur_hours = 0 ;
25             int cur_minutes = 0 ;
26             if(num < 0 || num > 10)
27             {
28                 return vs;
29             }
30
31             for( int i = 0 ; i < 2048; i++)
32             {
33                 if(bitcount(i)==num)
34                 {
35                     cur_hours = i/64;
36                     cur_minutes = i%64;
37                     if(cur_hours > 11 || cur_minutes > 59)
38                     {
39                         continue;
40                     }
41                     char buf_hour[3] = {0,};
42                     char buf_minute[3] = {0,};
43                     sprintf(buf_hour,"%d",cur_hours);
44                     sprintf(buf_minute,"%02d",cur_minutes);
45                     vs.push_back( buf_hour + string(":") + buf_minute);
46                 }
47             }
48             return vs;
49         }
50     };
```

```
51
52  /*
53  执行结果:
54  通过
55  显示详情
56  执行用时 :4 ms, 在所有 cpp 提交中击败了75.71% 的用户
57  内存消耗 :8.4 MB, 在所有 cpp 提交中击败了95.26%的用户
58  */
```

AlimyBreak
2019.11.30