

```

1  /*
2  格雷编码是一个二进制数字系统，在该系统中，两个连续的数值仅有一个位数的差异。
3
4  给定一个代表编码总位数的非负整数  $n$ ，打印其格雷编码序列。格雷编码序列必须以 0 开头。
5
6  示例 1:
7
8  输入: 2
9  输出: [0,1,3,2]
10 解释:
11  00 - 0
12  01 - 1
13  11 - 3
14  10 - 2
15
16 对于给定的  $n$ ，其格雷编码序列并不唯一。
17 例如，[0,2,3,1] 也是一个有效的格雷编码序列。
18
19 00 - 0
20 10 - 2
21 11 - 3
22 01 - 1
23
24 示例 2:
25
26 输入: 0
27 输出: [0]
28 解释: 我们定义格雷编码序列必须以 0 开头。
29         给定编码总位数为  $n$  的格雷编码序列，其长度为  $2^n$ 。当  $n = 0$  时，长度为  $2^0 = 1$ 。
30         因此，当  $n = 0$  时，其格雷编码序列为 [0]。
31
32 来源: 力扣 (LeetCode)
33 链接: https://leetcode-cn.com/problems/gray-code
34 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。
35 */

```

分析:

- **探索方法一(失败方法):**从左到右从右到左依次迭代取反某一位,在 $n = 2, 3$ 的时候碰巧通过,对于 $n \geq 4$ 时,总会在生成 $4(n - 1)$ 个数后陷入死循环且无法再生成更多(只验证到了 $n = 10$).

```

1  /*
2   $n = 4$ ,  $C = 12$ 
3  [0,1,3,7,15,11,9,8,10,14,6,2,0,1,3,7]
4   $n = 5$ ,  $C = 16$ 
5  [0,1,3,7,15,31,23,19,17,16,18,22,30,14,6,2,0,1,3,7,15,31,23,19,17,16,18,22,
6  30,14,6,2]
7   $n = 6$ ,  $C = 20$ 
8  [0,1,3,7,15,31,63,47,39,35,33,32,34,38,46,62,30,14,6,2,0,1,3,7,15,31,63,47,
9  39,35,33,32,34,38,46,62,30,14,6,2,0,1,3,7,15,31,63,47,39,35,33,32,34,38,46,
10 62,30,14,6,2,0,1,3,7]
11  $n = 7$ ,  $C = 24$ 

```

9 [0,1,3,7,15,31,63,127,95,79,71,67,65,64,66,70,78,94,126,62,30,14,6,2,0,1,3,
7,15,31,63,127,95,79,71,67,65,64,66,70,78,94,126,62,30,14,6,2,0,1,3,7,15,31,
63,127,95,79,71,67,65,64,66,70,78,94,126,62,30,14,6,2,0,1,3,7,15,31,63,127,
95,79,71,67,65,64,66,70,78,94,126,62,30,14,6,2,0,1,3,7,15,31,63,127,95,79,
71,67,65,64,66,70,78,94,126,62,30,14,6,2,0,1,3,7,15,31,63,127]

10 n = 8, c = 28

11 [0,1,3,7,15,31,63,127,255,191,159,143,135,131,129,128,130,134,142,158,190,2
54,126,62,30,14,6,2,0,1,3,7,15,31,63,127,255,191,159,143,135,131,129,128,13
0,134,142,158,190,254,126,62,30,14,6,2,0,1,3,7,15,31,63,127,255,191,159,14
,135,131,129,128,130,134,142,158,190,254,126,62,30,14,6,2,0,1,3,7,15,31,63,
127,255,191,159,143,135,131,129,128,130,134,142,158,190,254,126,62,30,14,6,
2,0,1,3,7,15,31,63,127,255,191,159,143,135,131,129,128,130,134,142,158,190,
254,126,62,30,14,6,2,0,1,3,7,15,31,63,127,255,191,159,143,135,131,129,128,1
30,134,142,158,190,254,126,62,30,14,6,2,0,1,3,7,15,31,63,127,255,191,159,14
3,135,131,129,128,130,134,142,158,190,254,126,62,30,14,6,2,0,1,3,7,15,31,63
,127,255,191,159,143,135,131,129,128,130,134,142,158,190,254,126,62,30,14,6
,2,0,1,3,7,15,31,63,127,255,191,159,143,135,131,129,128,130,134,142,158,190
,254,126,62,30,14,6,2,0,1,3,7]

12 n = 9, c = 32

13 [0,1,3,7,15,31,63,127,255,511,383,319,287,271,263,259,257,256,258,262,270,2
86,318,382,510,254,126,62,30,14,6,2,0,1,3,7,15,31,63,127,255,511,383,319,28
7,271,263,259,257,256,258,262,270,286,318,382,510,254,126,62,30,14,6,2,0,1,
3,7,15,31,63,127,255,511,383,319,287,271,263,259,257,256,258,262,270,286,31
8,382,510,254,126,62,30,14,6,2,0,1,3,7,15,31,63,127,255,511,383,319,287,271
,263,259,257,256,258,262,270,286,318,382,510,254,126,62,30,14,6,2,0,1,3,7,1
5,31,63,127,255,511,383,319,287,271,263,259,257,256,258,262,270,286,318,382
,510,254,126,62,30,14,6,2,0,1,3,7,15,31,63,127,255,511,383,319,287,271,263,
259,257,256,258,262,270,286,318,382,510,254,126,62,30,14,6,2,0,1,3,7,15,31,
63,127,255,511,383,319,287,271,263,259,257,256,258,262,270,286,318,382,510,
254,126,62,30,14,6,2,0,1,3,7,15,31,63,127,255,511,383,319,287,271,263,259,2
57,256,258,262,270,286,318,382,510,254,126,62,30,14,6,2,0,1,3,7,15,31,63,12
7,255,511,383,319,287,271,263,259,257,256,258,262,270,286,318,382,510,254,1
26,62,30,14,6,2,0,1,3,7,1...]

14 n = 10, c = 36

15 [0,1,3,7,15,31,63,127,255,511,1023,767,639,575,543,527,519,515,513,512,514,
518,526,542,574,638,766,1022,510,254,126,62,30,14,6,2,0,1,3,7,15,31,63,127,
255,511,1023,767,639,575,543,527,519,515,513,512,514,518,526,542,574,638,76
6,1022,510,254,126,62,30,14,6,2,0,1,3,7,15,31,63,127,255,511,1023,767,639,5
75,543,527,519,515,513,512,514,518,526,542,574,638,766,1022,510,254,126,62,
30,14,6,2,0,1,3,7,15,31,63,127,255,511,1023,767,639,575,543,527,519,515,513
,512,514,518,526,542,574,638,766,1022,510,254,126,62,30,14,6,2,0,1,3,7,15,3
1,63,127,255,511,1023,767,639,575,543,527,519,515,513,512,514,518,526,542,5
74,638,766,1022,510,254,126,62,30,14,6,2,0,1,3,7,15,31,63,127,255,511,1023,
767,639,575,543,527,519,515,513,512,514,518,526,542,574,638,766,1022,510,25
4,126,62,30,14,6,2,0,1,3,7,15,31,63,127,255,511,1023,767,639,575,543,527,51
9,515,513,512,514,518,526,542,574,638,766,1022,510,254,126,62,30,14,6,2,0,1
,3,7,15,31,63,127,255,511,1023,767,639,575,543,527,519,515,513,512,514,518,
526,542,574,638,766,1022,...]

16 */

17

- **方法二:**格雷编码的生成过程, 好吧, 我背的答案, $vi[i] = i \text{ xor } (i/2)$.
- **方法三:**递归算法, 致谢 [ygh578890840](#)
 - 若已知 $n - 1$ 的格雷编码集合, 在前面加一个 0, 然后把 $n - 1$ 的集合倒转在前面加一个 1, 就得到了 n 的格雷编码集合.

- 翻译一下：把前(n-1)层的所有格雷码都按(n-1)位取反，然后把第n位置为1，依次压入返回数组

方法一:C++_失败方法

```
1
2 class solution
3 {
4     public:
5         vector<int> grayCode(int n)
6         {
7             vector<int> vi ;
8             int cur_v = 0 ;
9             int i = 0 ;
10            int sum_count = pow(2,n) ;
11            int cur_count = 0 ;
12            int cur_idx = 0 ;
13            int direction = 0 ;
14            if(n >=0)
15            {
16                vi.push_back(cur_v) ;
17                cur_count = 1 ;
18                cur_idx = 0 ;
19                while(cur_count<sum_count)
20                {
21                    cur_v ^= (0x01 << cur_idx);
22                    if(cur_idx == n-1)
23                    {
24                        direction = 0; /* left->right*/
25                        cur_idx--;
26                    }
27                    else if(cur_idx == 0)
28                    {
29                        direction = 1;
30                        cur_idx++;
31                    }
32                }
33                else if(direction==0)
34                {
35                    cur_idx--;
36                }
37                else
38                {
39                    cur_idx++;
40                }
41
42                vi.push_back(cur_v) ;
43                cur_count++;
44            }
45        }
46        return vi;
47    }
48 };
49
50 /*
51 在 n >= 4,无法生成满足要求的序列。
```

```
52  */
53
```

方法三:C++_递归算法

```
1  class Solution
2  {
3      public:
4          vector<int> grayCode(int n)
5          {
6              if(n == 0)
7              {
8                  return {0};
9              }
10             vector<int> vi = grayCode(n-1) ;
11             int i = 0 ;
12             /*n层共有 2^n 个格雷码，前(n-1)层共有 2^(n-1)个格雷码*/
13             for( i = pow(2,n-1)-1; i >= 0 ; i--)
14             {
15                 vi.push_back(vi[i] | (0x01 << (n-1)));
16             }
17             return vi;
18         }
19     };
20     /*
21     执行结果:
22     通过
23     显示详情
24     执行用时 :12 ms，在所有 cpp 提交中击败了18.56% 的用户
25     内存消耗 :8.7 MB，在所有 cpp 提交中击败了20.23%的用户
26     */
```