89.格雷编码-【回溯算法超时】【位运算】

89. 格雷编码

难度 中等 282

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。

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。即使有多个不同答案，你也只需要返回其中一种。

格雷编码序列必须以 0 开头。

2. 第二个点，格雷编码如何形成。同样以n = 2解析。
    - 公式为 i ^ i/2 (i >> 1)
    - 0 ^ 0 = 00 ^ 00 = 00 = 0 计算(i/2): 0/2 = 0
    - 1 ^ 0 = 01 ^ 00 = 01 = 1 计算(i/2): 1/2 = 0
    - 2 ^ 1 = 10 ^ 01 = 11 = 3 计算(i/2): 2/2 = 1
    - 3 ^ 1 = 11 ^ 01 = 10 = 2 计算(i/2): 3/2 = 1

```
class Solution {
public:
    vector<int> grayCode(int n) {
        vector<int> ans;
        int powN = 1 << n;
        for(int i = 0; i < powN; ++i)
            ans.push_back(i^i>>1);
        return ans;
    }
};
```

回溯超时【用例16】

```
class Solution {
    private:
    //unordered_set<int>res;
    vector<int>ans;
    vector<int>res;
    int sum,max_sum,flag;

    int change_type(vector<int>&codes){
        int num=0;
        for(int i=0;i<codes.size();i++){
            num*=2;
            num+=codes[i];
        }
        return num;
    }

    bool backtracking(vector<int>&codes){
        //终止条件
        if(flag)return false;
        if(sum==max_sum&&!flag){
            //unordered_set<int>::iterator it;
            //for (it = res.begin(); it != res.end(); ++it)
            //ans.push_back(*it);
            flag=1;
            for(auto tmp:ans){
                res.push_back(tmp);
            }
            return true;
        }
        //if(res.count(change_type(codes)))return false;
        //res.insert(change_type(codes));
        int num=change_type(codes);
        for(int i=0;i<ans.size();i++){
            if(ans[i]==num)return false;
        }
        ans.push_back(num);
        sum++;
```

```
        for(int i=0;i<codes.size();i++){
            codes[i]=!codes[i];
            backtracking(codes);
            codes[i]=!codes[i];
        }
        ans.pop_back();
        return false;
    }

public:
    vector<int> grayCode(int n) {
        //穷举所有可能
        //回溯算法
        vector<int>codes(n,0);
        sum=0;max_sum=pow(2,n);flag=0;
        //res.insert(0);//初始化
        backtracking(codes);
        return res;

    }
```