# 题目

中心对称数是指一个数字在旋转了 180 度之后看起来依旧相同的数字（或者上下颠倒地看）。

找到所有长度为 n 的中心对称数。

示例 :

输入: n = 2

输出: ["11","69","88","96"]

# 分析

class Solution {

public:

vector<string> helper(int n) {

if (n == 1) {

return vector<string>{"0", "1", "8"};

}

if (n == 2) {

return vector<string>{"11", "69", "88", "96", "00"};

}

vector<string> subs = helper(n-2);

vector<string> ans;

for (auto sub: subs) {

ans.push\_back("0"+sub+"0");

ans.push\_back("1"+sub+"1");

ans.push\_back("6"+sub+"9");

ans.push\_back("9"+sub+"6");

ans.push\_back("8"+sub+"8");

}

return ans;

}

vector<string> findStrobogrammatic(int n) {

if (n == 1) {

return vector<string>{"0", "1", "8"};

}

if (n == 2) {

return vector<string>{"11", "69", "88", "96"};

}

vector<string> subs = helper(n-2);

vector<string> ans;

for (auto sub: subs) {

ans.push\_back("1"+sub+"1");

ans.push\_back("6"+sub+"9");

ans.push\_back("9"+sub+"6");

ans.push\_back("8"+sub+"8");

}

return ans;

}

};

另一种写法：

class Solution

{

public:

int n;

vector<pair<char,char>> pairs;

vector<string> dfs(int x)

{

if (x == 0)

return {""};

else if(x == 1)

return vector<string>{"0", "1", "8"};

vector<string> res{};

for (string num: dfs(x - 2))

{

for(auto [a,b] : pairs)

{

res.push\_back(a + num + b); //两边加的办法

}

if (x != n)

res.push\_back('0' + num + '0'); //0要慎重

}

return res;

}

vector<string> findStrobogrammatic(int n)

{

this->n = n;

pairs = vector<pair<char,char>> {{'1', '1'}, {'8','8'}, {'6','9'}, {'9','6'} };

return dfs(n);

}

};