

2020 年 8 月 4 日

1.P1219 [USACO1.5]八皇后 Checker Challenge

算法思路：

dfs+回溯。按行来找合适的列，queen记录每一行棋子所在的列，c记录已被占用的列，r记录已被占用的右对角线，l记录左已被占用的左对角线（左右对角线分别是相对于当前点而言，往左上角 或者 往右上角移位），发现规律： 在同一条左对角线的，所在列-所在行的值相等，这个值作为l的下标，记录这条对角线已被占用，注意数组越界所以 +15 偏移量；在同一条右对角线的，所在行+所在列的值相等，这个值作为r的下标，记录这条对角线已被占用。

代码：

```
#include <iostream>

using namespace std;

#define N 30
int queen[N] = {0};
int res = 0, n, cnt = 0;
int c[N] = {0}, r[N] = {0}, l[N] = {0};

void dfs(int nrow, int ncol){

    queen[nrow] = ncol;
    //cout << cnt << endl;
    if(cnt == n){
        res++;
        if(res <= 3){
            for(int i = 1; i < n; i++){
                cout << queen[i] << " ";
            }
            cout << queen[n] << endl;
        }
        return;
    }

    for(int i = 1; i <= n; i++){
        int sum = i + nrow + 1;
        int diff = i - nrow - 1 + 15;
        //cout << sum << " " << diff << endl;
        if( c[i] != 1 && l[diff] != 1 && r[sum] != 1){ //不同列，不同对角
```

线

```

        cnt++;
        c[i] = 1;
        l[diff] = 1;
        r[sum] = 1;
        dfs(nrow+1,i);
        c[i] = 0;
        l[diff] = 0;
        r[sum] = 0;
        cnt--;
    }
}

int main(){

    cin>>n;
    for(int i = 1; i <= n; i++){
        cnt++;
        c[i] = 1;
        r[1+i] = 1;
        l[i-1+15] = 1;
        dfs(1,i);
        cnt--;
        c[i] = 0;
        r[1+i] = 0;
        l[i-1+15] = 0;
    }
    cout<<res;
    return 0;
}

```

Accepted截图：



 **wongsiyoung**
 所属题目 [P1219 \[USACO1.5\]八皇后 Checker Challenge](#)
 评测状态 **Accepted**
 评测分数 **100**
 提交时间 2020-08-04 13:15:06

备注：

1. 一开始只考虑当前找的列不要同上一行的值在同一列和同一条对角线，然后才想到是所有的之前已经填过的不能在同一列和同一条对角线，然后快快的看了姐姐的代码（没看仔细~），看到姐姐前面是用了三个数组来记录，然后就试着再做一下~

2. 增加三个数组来记录同列和同对角线的时候，左对角线的越界问题，一开始我是用了一一定是（大数-小数），而不是列-行，所以会导致两条对角线平行的话，会被当成同一条对角线，然后结果就是 0 了，没有解~
这才理解了姐姐为啥右斜线要+15~