

《算法分析与设计》第 4 次作业^{*}

姓名：谈金翰 学号：71118314

2020/4/11

算法分析题

题目：符号三角形

下图所示的三角形中，有14个“+”和14个“-”。2个同号下面是+，两个异号下面是-。

```
+ + - + - + +
+ - - - - +
- + + + -
- + + -
- + -
- -
+
```

在一般情况下，符号三角形的第一行有 n 个符号。符号三角形问题，要求对于给定的 n ，计算有多少个不同的符号三角形，使其所含的“+”和“-”相同。

请设计一个回溯算法求解符号三角形问题

答：

算法思路：

只要确定了第一行的符号，则后面的符号也跟着确定了，因此可将问题简化成第一行符号的组合问题

第一行符号，只有两种情况，非+即-，因此可用0表示-，1表示+

此问题简化成求一个 n 位二进制数的所有组合

之后只需根据第一行的符号，计算+-号的数量是否相等即可，

相等即输出，不等就回溯

可以采用的优化思想：

- 1) 所有符号需为偶数，因此如果总数为奇数直接跳过
- 2) 当任意一个符号的数量超过总数的一半时，跳过

伪代码如下：

输入 n

输出 符号三角形总数，所有符号三角形

1. $m \leftarrow (n+1)n/2$

2. if(m 为奇数)

3. then end

4. else

5. change(0)

change(j):

6. if($j==n$)

7. then 判断是否满足符号三角形条件

8. else

9. $signal[j]=0$

10. change($j+1$)

11. $signal[j]=1$ //回溯

12. change($j+1$)

13. endif

```
1 #include<iostream>
   using namespace std;
3
   int* signal;//符号三角形数组
5 int m;//总符号数m=(n+1)n/2
   int n;//输入值
7 int total;//总的符号三角形个数
   int c1;//+数量
9 int c2;//-数量

11 void change( int );//求出n位二进制数
   void print();//打印三角形
13 void judge();//判断是否为符号三角形

15 int main() {
   cin >> n;
17 while (n > 0) {
   total = 0;//记录总数
19 m = n * (n + 1) / 2;//总的符号数量
   signal = new int[m]; //符号三角形数组
21 // fill(signal, signal+m, 0);
```

```

23     if (m % 2 == 1)
24         cout << "n=" << n << ":no signal triangle!" << endl;
25     else {
26         change(0);
27         if (total == 0)
28             cout << "n=" << n << ":no signal triangle!" << endl;
29             cout << "一共" << total << "个符号三角形" << endl;
30     }
31     cin >> n;
32 }
33 }

35 void change(int j) {
36     if (j == n)
37         judge();
38     else {
39         signal[j] = 0;
40         change(j + 1);
41         signal[j] = 1; //回溯
42         change(j + 1);
43     }
44 }

45 void judge() {
46     c1 = c2 = 0;
47     int k = 0;
48     int row = 0;
49     int s = n;
50     for (; k < n; k++) //计算第一行正负号个数
51     {
52         if (signal[k] == 1) c1++;
53         else c2++;
54     } //计算后面的+ -号数量
55     for (int i = 0; k < m; i++) {
56         if (i == s - 1) { //换行
57             i = s;
58             row++;
59             s += n - row;
60         }
61         if (signal[i] == signal[i + 1]) { //正好
62             signal[k++] = 1;
63             c1++;
64         }
65         else { //负号
66             signal[k++] = 0;

```

```

        c2++;
69     }
    if (c1 > m / 2 || c2 > m / 2) // 如果超过总数一半则直接结束
71     return;
}

73
if (c1==c2) {
75     total++;
    print();
77 }
}

79 }

81 void print() {
    int s = n;
83     int k = 0;
    cout << "第" << total << "种符号三角形为：" << endl;
85     for (int i = 0; i < m; i++) {
        if (signal[i] == 1)
87             cout << "+";
        else
89             cout << "-";
        cout << " ";
91     if (i == s-1) { // 换行
        cout << endl;
93         k = k + 1;
        s += n - k;
95         for (int u = 0; u < k; u++)
            cout << " ";
97     }
    }
99     cout << endl;
}

```

第409种符号三角形为:

```
+ + + + + - + - - - - +
+ + + + - - - + + + -
+ + + - + + - + + -
+ + - - + - - + -
+ - + - - + - -
- - - + - - +
+ + - - + -
+ - + - -
- - - +
+ + -
+ -
-
```

第410种符号三角形为:

```
+ + + + + - + + - + + -
+ + + + - - + - - + -
+ + + - + - - + - -
+ + - - - + - - +
+ - + + - - + -
- - + - + - -
+ - - - - +
- + + + -
- + + -
- + -
- -
+
```

一共410个符号三角形

6

n=6:no signal triangle!

28

一共11191932个符号三角形

0

D:\Project\VS\算法\特工和核电
按任意键关闭此窗口...

实验测试结果如图: