

# “+”、“-”号构成的阵列 The Matrixes Consisted of “+” and “-”

罗海鹏      黎贞崇  
Lou Haipeng      Li Zhencong

苏芳来  
Su Fanglai

(广西科学院 南宁 530031)  
(Guangxi Academy of Sciences,  
Nanning, 530031)

(广西电化教育馆 南宁 530023)  
(Audio-Visual Educational  
Centre of Guangxi, Nanning, 530023)

**摘要** 寻找“+”、“-”号个数相等的阵列问题是尚未完全解决的问题。给出它的计算机算法,并具体地算出了第一行有 $N_1(3 \leq N_1 \leq 36)$ 个“+”、“-”号的解的例子,算出了第一行有 $N_2(3 \leq N_2 \leq 24)$ 个“+”、“-”号的解的个数。

**关键词** “+”、“-”号阵列 计算机算法

**Abstract** Finding the matrixes which consist of same individuals of “+” and “-” is also a problem that has not been solved completely. An algorithms for it by computer was pointed out, and both an sample with  $N_1(3 \leq N_1 \leq 36)$  individuals of “+” and “-” in row 1 and the  $N_2(3 \leq N_2 \leq 24)$  individuals of “+” and “-” in row 1 were computed.

**Key words** “+”、“-” matrix, algorithms by computer

中图法分类号 O 241.6; TP 301.6

右图是由14个“+”号和14个“-”号组成的阵列。它形成的规则是这样的:第一行的7个“+”、“-”号是任意的,以下各行每个符号都要看它左上方、右上方的符号,如果同号则它为“+”号,如果异号则它为“-”号。

显而易见,如果第一行有 $N$ 个“+”、“-”号,则这个阵列共有 $N(N+1)/2$ 个“+”、“-”号,如果我们要研究阵列中“+”号个数与“-”号个数相等的情况,一定要有 $N = 3, 4, 7, 8, 11, 12, \dots$ ,即 $N = 4L - 1$ 或 $4L(L = 1, 2, 3, \dots)$ 。

找出符合上述规则的“+”、“-”号阵列的通解是一个仍未解决的问题<sup>[1]</sup>。

+	+	-	+	-	+	+
+	-	-	-	-	-	+
-	+	+	+	-	-	-
-	+	+	-	-	-	-
-	+	-	-	-	-	-
-	-	-	-	-	-	-
+	-	-	-	-	-	-

## 1 算法描述与程序

我们通过计算机来研究这个问题。

### 1.1 算法描述

- 1)  $N$  个元素的一个字符数组每个单元都装入“+”;
- 2) 按照二进制的改变数值和进位方式,从高位到低位改变“+”、“-”号,每一次改变,都形成一个新的阵列的第一行;
- 3) 产生阵列的其他行;
- 4) 统计“+”、“-”号个数,如相等则输出这个阵列,结束;如不等,则分两种情形:a) 第一行已全部是“-”号,则输出“THE MATRIX NOT EXIST.”结束;b) 第一行不是全部“-”号,则转2)。

### 1.2 C语言程序清单

```
main()
{
    int m,n;
    long k,i,x,y,s;
    char AY[20],BY[20];

    k=0;
    s=524288L;
    for(m=1;m<19;++m)
        AY[m]='+';
    for(i=1;i<=s;++i)
    {
        for(m=1;AY[m]== '-'&& m<=19;++m)
            AY[m]='+';
        AY[m]='-';
        x=0;y=0;

        for(m=1;m<=19;++m)
            BY[m]=AY[m];
        for(m=1;m<=19;++m)
        {
            for(n=1;n<=20-m;++n)
            {
                if(BY[n]== '+')
                    ++x;
                else
                    ++y;
                if(BY[n]==BY[n+1])
                    BY[n]='+';
                else
                    BY[n]='-'
            }
        }
    }
    if(x==y)
    {
```

```

++k;
for(m=1;m<=19;++m)
    printf("%c",AY[m]);
    printf("\n");
}
printf("%1d\n",k);
}

```

注：S=524288L,L 为定义长整数

### 2 “+”、“-”号相等的阵列的例子

通过计算机计算,我们找出了很多“+”、“-”号相等的阵列,下面对 N=4L-1 或 4L(L=1,2,...,8)各举出 1 例子。

#### 2.1 N=3 的例子

```

-++
-+
-

```

#### 2.3 N=7 的例子

```

-+-++
-+--+
--+-+
+---
-++
-+
-

```

#### 2.4 N=8 的例子

```

-+-++++
+----+++
-+-++
-+--+
--+-
+--
--+
-

```

#### 2.2 N=4 的例子

```

+-++
--+
+-
-

```

以下仅列出阵列的第一行。

#### 2.5 N=11 的例子

-+-++-++++

#### 2.6 N=12 的例子

-+-++-++-++++

#### 2.7 N=15 的例子

--+-+--++-+++++

#### 2.8 N=16 的例子

+---+---+---+++++

#### 2.9 N=19 的例子

--++-+-+---++-+++++

#### 2.10 N=20 的例子

+---++-+-+---++-+++++

#### 2.11 N=23 的例子

-+-++-+---+---++-+++++

#### 2.12 N=24 的例子

+---++-+-+---++-+++++

#### 2.13 N=27 的例子

++-+---+---+---++-+++++

