

Google 面试题——更改序列顺序问题解答

作者：余祖波(livelylittlefish@gmail.com)

Blog: <http://blog.csdn.net/livelylittlefish>, <http://www.abo321.org>

Content

1. 每次交换 1 个
2. 交换次数变化
3. 交换次数变化
4. 不需移动，通过交换完成

问题：输入 $a_1, a_2, \dots, a_n, b_1, b_2, \dots, b_n$ ，如何在 $O(n)$ 的时间，用 $O(1)$ 的空间，将这个序列顺序改为 $a_1, b_1, a_2, b_2, \dots, a_n, b_n$ 。

1. 每次交换 1 个

如下表所示，颜色表示每次要交换的元素

$N=10$

| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | n+1 | n+2 | n+3 | n+4 | n+5 | n+6 | n+7 | n+8 | n+9 | n+10 | | |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|------|------|------|------|------|------|------|------|------|------|-------------------------|----------------------------|
| n-9 | n-8 | n-7 | n-6 | n-5 | n-4 | n-3 | n-2 | n-1 | n | 2n-9 | 2n-8 | 2n-7 | 2n-6 | 2n-5 | 2n-4 | 2n-3 | 2n-2 | 2n-1 | 2n | 位置 | |
| a1 | a2 | a3 | a4 | a5 | a6 | a7 | a8 | a9 | a10 | b1 | b2 | b3 | b4 | b5 | b6 | b7 | b8 | b9 | b10 | $2 \leftrightarrow n+1$ | $2n-1 \leftrightarrow n$ |
| a1 | b1 | a3 | a4 | a5 | a6 | a7 | a8 | a9 | b9 | a2 | b2 | b3 | b4 | b5 | b6 | b7 | b8 | a10 | b10 | $3 \leftrightarrow n+1$ | $2n-2 \leftrightarrow n$ |
| a1 | b1 | a2 | a4 | a5 | a6 | a7 | a8 | a9 | b8 | a3 | b2 | b3 | b4 | b5 | b6 | b7 | b9 | a10 | b10 | $4 \leftrightarrow n+2$ | $2n-3 \leftrightarrow n-1$ |
| a1 | b1 | a2 | b2 | a5 | a6 | a7 | a8 | b7 | b8 | a3 | a4 | b3 | b4 | b5 | b6 | a9 | b9 | a10 | b10 | $5 \leftrightarrow n+1$ | $2n-4 \leftrightarrow n$ |
| a1 | b1 | a2 | b2 | a3 | a6 | a7 | a8 | b7 | b6 | a5 | a4 | b3 | b4 | b5 | b8 | a9 | b9 | a10 | b10 | $6 \leftrightarrow n+3$ | $2n-5 \leftrightarrow n-2$ |
| a1 | b1 | a2 | b2 | a3 | b3 | a7 | b5 | b7 | b6 | a5 | a4 | a6 | b4 | a8 | b8 | a9 | b9 | a10 | b10 | $7 \leftrightarrow n+2$ | $2n-6 \leftrightarrow n-1$ |
| a1 | b1 | a2 | b2 | a3 | b3 | a4 | b5 | b4 | b6 | a5 | a7 | a6 | b7 | a8 | b8 | a9 | b9 | a10 | b10 | $8 \leftrightarrow n-1$ | $2n-7 \leftrightarrow n-2$ |
| a1 | b1 | a2 | b2 | a3 | b3 | a4 | b4 | b5 | b6 | a5 | a6 | a7 | b7 | a8 | b8 | a9 | b9 | a10 | b10 | | |

该方案当 n 越来越大时，交换位置毫无规律，编程不能实现

2. 交换次数变化

考虑到第一次交换后实际上成功配对的有 4 对，头尾各 1 对，中间对称位置各 1 对，其位置尚需交换，如下所示。

$N=5$

| 1 | 2 | 3 | 4 | 5 | n+1 | n+2 | n+3 | n+4 | n+5 | | | |
|-----|-----|-----|-----|----|------|------|------|------|-----|-------------------------|--------------------------|---|
| n-4 | n-3 | n-2 | n-1 | n | 2n-4 | 2n-3 | 2n-2 | 2n-1 | 2n | | | |
| a1 | a2 | a3 | a4 | a5 | b1 | b2 | b3 | b4 | b5 | $2 \leftrightarrow n+1$ | $2n-1 \leftrightarrow n$ | 1 |
| a1 | b1 | a3 | a4 | b4 | a2 | b2 | b3 | a5 | b5 | 交换 | | |
| a1 | b1 | a3 | a2 | b2 | a4 | b4 | b3 | a5 | b5 | 左移 | | |
| a1 | b1 | a2 | b2 | a3 | b3 | a4 | b4 | a5 | b5 | | | |

$N=10$

| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | n+1 | n+2 | n+3 | n+4 | n+5 | n+6 | n+7 | n+8 | n+9 | n+10 | | | |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|------|------|------|------|------|------|------|------|------|------|-------------------------|--------------------------|---|
| n-9 | n-8 | n-7 | n-6 | n-5 | n-4 | n-3 | n-2 | n-1 | n | 2n-9 | 2n-8 | 2n-7 | 2n-6 | 2n-5 | 2n-4 | 2n-3 | 2n-2 | 2n-1 | 2n | | | |
| a1 | a2 | a3 | a4 | a5 | a6 | a7 | a8 | a9 | a10 | b1 | b2 | b3 | b4 | b5 | b6 | b7 | b8 | b9 | b10 | $2 \leftrightarrow n+1$ | $2n-1 \leftrightarrow n$ | 1 |
| a1 | b1 | a3 | a4 | a5 | a6 | a7 | a8 | a9 | b9 | a2 | b2 | b3 | b4 | b5 | b6 | b7 | b8 | a10 | b10 | $3 \leftrightarrow n+1$ | $2n-2 \leftrightarrow n$ | 2 |
| a1 | b1 | a2 | b2 | a5 | a6 | a7 | a8 | b7 | b8 | a3 | a4 | b3 | b4 | b5 | b6 | a9 | b9 | a10 | b10 | $5 \leftrightarrow n+1$ | $2n-4 \leftrightarrow n$ | 4 |
| a1 | b1 | a2 | b2 | a5 | a6 | a7 | b7 | a8 | b8 | a3 | b3 | a4 | b4 | b5 | b6 | a9 | b9 | a10 | b10 | 交换 | | |
| a1 | b1 | a2 | b2 | a5 | a6 | a7 | b3 | a4 | b4 | a7 | b7 | a8 | b8 | b5 | b6 | a9 | b9 | a10 | b10 | 互换 | 左移 | |
| a1 | b1 | a2 | b2 | a3 | b3 | a4 | b4 | a5 | a6 | b5 | b6 | a7 | b7 | a8 | b8 | a9 | b9 | a10 | b10 | | | |

算法思想：

以 $N=9$ 为例(中间的竖表示中间位置)：

a1 a2 a3 a4 a5 a6 a7 a8 a9 | b1 b2 b3 b4 b5 b6 b7 b8 b9

头尾的元素不需任何操作

1. 左边从位置 $left=2$ 开始，右边从位置 $n+1$ 开始，向右交换 $count=1$ 个元素，即 a2, b1 交换

右边从位置 $right=2n-1$ 开始，左边从位置 n 开始，向左交换 $count=1$ 个元素，即 b8, a9 交换

序列变为：

a1 b1 a3 a4 a5 a6 a7 a8 b8 | a2 b2 b3 b4 b5 b6 b7 a9 b9

故已经成功放好位置的有(a1, b1), (a9, b9)

其中(a8, b8), (a2, b2) 也配对，只需将其交换到相应的位置即可

2. 左边从位置 $left=3$ 开始，右边从位置 $n+1$ 开始，向右交换 $count=2$ 个元素，即 a3 a4 和 a2 b2 交换

右边从位置 $right=2n-2$ 开始，左边从位置 n 开始，向左交换 $count=2$ 个元素，即 b6 b7 和 a8 b8 交换

序列变为：
a1 b1 a2 b2 a5 a6 a7 b6 b7 | a3 a4 b3 b4 b5 a8 b8 a9 b9

故又成功放好位置的有(a2, b2), (a8, b8)

3. 左边从位置 `left=5` 开始，右边从位置 `n` 开始，已不能满足交换 `count=4` 个元素的要求，故退出循环

从中间位置对称交换 `count=count/2=2` 个元素

序列变为：

a1 b1 a2 b2 a5 a6 a7 a3 a4 | b6 b7 b3 b4 b5 a8 b8 a9 b9

4. 左边从位置 `left=5` 开始的 `size=n-left+1=5` 个元素循环左移 `x=n-left-count+1=3` 位

右边从位置 `n+1` 开始的 `size=5` 个元素循环右移 3 位

序列变为：

a1 b1 a2 b2 a3 a4 a5 a6 a7 | b3 b4 b5 b6 b7 a8 b8 a9 b9

5. 至此，将原来的序列缩小为：

a3 a4 a5 a6 a7 | b3 b4 b5 b6 b7

对该序列进行上述操作，直到所有元素都放到正确的位置

为方便比较，序列变化如下：

| | | | | | | | | | | | | | | | | | | |
|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|--|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|
| <u>a1</u> | <u>a2</u> | <u>a3</u> | <u>a4</u> | <u>a5</u> | <u>a6</u> | <u>a7</u> | <u>a8</u> | <u>a9</u> | | <u>b1</u> | <u>b2</u> | <u>b3</u> | <u>b4</u> | <u>b5</u> | <u>b6</u> | <u>b7</u> | <u>b8</u> | <u>b9</u> |
| <u>a1</u> | <u>b1</u> | <u>a3</u> | <u>a4</u> | <u>a5</u> | <u>a6</u> | <u>a7</u> | <u>a8</u> | <u>b8</u> | | <u>a2</u> | <u>b2</u> | <u>b3</u> | <u>b4</u> | <u>b5</u> | <u>b6</u> | <u>b7</u> | <u>a9</u> | <u>b9</u> |
| <u>a1</u> | <u>b1</u> | <u>a2</u> | <u>b2</u> | <u>a5</u> | <u>a6</u> | <u>a7</u> | <u>b6</u> | <u>b7</u> | | <u>a3</u> | <u>a4</u> | <u>b3</u> | <u>b4</u> | <u>b5</u> | <u>a8</u> | <u>b8</u> | <u>a9</u> | <u>b9</u> |
| <u>a1</u> | <u>b1</u> | <u>a2</u> | <u>b2</u> | <u>a3</u> | <u>a4</u> | <u>a5</u> | <u>a6</u> | <u>a7</u> | | <u>b3</u> | <u>b4</u> | <u>b5</u> | <u>b6</u> | <u>b7</u> | <u>a8</u> | <u>b8</u> | <u>a9</u> | <u>b9</u> |

源程序如下：

```

/*****
* 输入a1,a2,...,an,b1,b2,...,bn
* 在O(n)的时间,O(1)的空间
* 将这个序列顺序改为a1,b1,a2,b2,a3,b3,...,an,bn
*****/

```

```

#include <stdio.h>
#include <string.h>
#include <CONIO.H>

```

```

#define MAXSIZE 2*1000

```

```

int exchangetimes=0; //交换次数
int movetimes=0;    //移动次数

```

//交换两个数据

```

void swap(int *x,int *y)

```

```

{
    int t;
    t=*x;
    *x=*y;
    *y=t;

    exchangetimes++;
}

```

//将数组中的元素循环左移x位,n为数组的元素个数

```

void rotate_left(int a[],int n,int x)

```

```

{
    int t;
    for(int i=0;i<x;i++)
    {
        t=a[0];
        for(int j=0;j<n-1;j++)
            a[j]=a[j+1];
        a[n-1]=t;
    }

    movetimes+=n*x;
}

```

//将数组中的元素循环右移x位,n为数组的元素个数

```

void rotate_right(int a[],int n,int x)

```

```

{
    int t;
    for(int i=0;i<x;i++)
    {
        t=a[n-1];
        for(int j=n-1;j>0;j--)
            a[j]=a[j-1];
        a[0]=t;
    }
}

```

```

    movetimes+=n*x;
}

//按要求交换序列(假设元素从下标1开始存放)
void exchange(int a[],int m)
{
    int n=m/2;

    if(n==1)        //a1,b1 ==>不需交换
        return;
    else if(n==2)    //a1,a2,b1,b2 ==>只需交换中间的两个数据
    {
        swap(a+1,a+2);
        return;
    }

    int done;        //已经处理的数据个数
    int left;         //左边开始交换的位置
    int right;        //右边开始交换的位置
    int count;        //每次交换的数据个数
    int lefta;         //左边未处理的a个数
    int notmatch;     //左边未匹配的a个数

    //初始化
    done=1;
    left=1;           //左边从位置1开始向右交换
    right=2*n-2;      //右边从位置2*n-2开始向左交换
    count=1;          //每次交换count个数据
    lefta=notmatch=n-1;

    do
    {
        //交换连续的count个元素
        for(int j=0;j<count;j++)
            swap(a+left+j,a+n+j);

        //若notmatch=0,该交换不能进行,否则为重复交换
        if(notmatch>0)
        {
            for(j=0;j<count;j++)
                swap(a+right-j,a+n-1-j);
        }
        else
            break;

        //交换后将其调整为要求的序列
        if(count>=4)
        {
            exchange(a+left,count);
            exchange(a+right-count+1,count);
        }

        //重新调整各变量
        done+=count;
        lefta=n-done-count;
        notmatch=lefta-count;

        left=left+count;
        right=right-count;

        count*=2;

        if(notmatch<count)
            break;
    }while(1);

    if(notmatch<count)
    {
        count/=2;
    }
}

```

```

//由中间对称交换count个数据
for(int j=0;j<count;j++)
    swap(a+n-count+j,a+n+j);

int size=n-left;

//左边的数据循环左移lefta位
rotate_left(a+left,size,lefta);

//右边的数据循环右移lefta位
rotate_right(a+n,size,lefta);

int newm=2*size;
//递归调用
exchange(a+left,newm);
}
}

//显示菜单
void show_menu()
{
    printf("-----\n");
    printf("input command to test the program\n");
    printf("    i or I : input n to test\n");
    printf("    t or T : test program\n");
    printf("    q or Q : quit\n");
    printf("-----\n");
    printf("$ input command >");
}

//显示数据
void display(int a[],int n)
{
    for(int i=0; i<n; i++)
        printf("%3d",a[i]);
    printf("\n");
}

//检查交换是否正确
bool check(int a[],int n)
{
    int i;

    for(i=0;i<n-2;i+=2)
    {
        if(a[i]!=i/2+1)
            return false;
    }

    for(i=1;i<n-1;i+=2)
    {
        if(a[i]!=(n+i+1)/2)
            return false;
    }

    return true;
}

void main()
{
    int a[MAXSIZE];
    int n;
    char sinput[10];

    show_menu();

    scanf("%s",sinput);
    while(stricmp(sinput,"q")!=0)
    {
        if(stricmp(sinput,"i")==0)
        {
            printf("    please input n:");

```

```

scanf("%d",&n);

//假设数组的第0个元素为0,且不对其进行操作
//且假设数组中的数据为1,2,3,...,n,n+1,n+2,...,2n
for(int i=0; i<2*n; i++) //初始化
    a[i]=i+1;

display(a,2*n);

exchangetimes=0;
movetimes=0;

//交换
exchange(a,2*n);
display(a,2*n);

printf("\n  exchange times: %d\n",exchangetimes);
printf("  move times: %d\n",movetimes);
}
else if(stricmp(sinput,"t")==0)
{
    int n1,n2;
    printf("  please input the begin number:");
    scanf("%d",&n1);
    printf("  please input the end number:");
    scanf("%d",&n2);
    printf("  press any key to start ...\n\n");
    getch();

    for(int i=n1; i<=n2; i++)
    {
        //初始化
        for(int j=0; j<2*i; j++)
            a[j]=j+1;

        exchangetimes=0;
        movetimes=0;
        exchange(a,2*i);

        if(check(a,2*i))
            //printf("  n=%d ... ok!\n",i);
            printf("  n=%d ... ok!    exchange times: %d    move times: %d\n",i,exchangetimes,movetimes);
        else
            printf("  n=%d ... wrong!\n",i);
    }
    printf("\n");
}
//输入命令
printf("$ input command >");
scanf("%s",sinput);
}
}
}

```

运行结果如下:

```

C:\Documents and Settings\yuzubo\桌面\test\Debug\exchange1.exe
-----
input command to test the program
i or I : input n to test
t or T : test program
q or Q : quit
-----
$ input command >i
please input n:7
1 2 3 4 5 6 7 8 9 10 11 12 13 14
1 8 2 9 3 10 4 11 5 12 6 13 7 14

exchange times: 11
move times: 6

$ input command >i
please input n:9
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18
1 10 2 11 3 12 4 13 5 14 6 15 7 16 8 17 9 18

exchange times: 14
move times: 42

$ input command >

```

```

C:\Documents and Settings\yuzubo\桌面\test\Debug\exchange1.exe
input command to test the program
i or I : input n to test
t or T : test program
q or Q : quit

$ input command >t
please input the begin number:11
please input the end number:20
press any key to start ...

n=11 ... ok!    exchange times: 19    move times: 76
n=12 ... ok!    exchange times: 24    move times: 4
n=13 ... ok!    exchange times: 26    move times: 22
n=14 ... ok!    exchange times: 29    move times: 24
n=15 ... ok!    exchange times: 31    move times: 48
n=16 ... ok!    exchange times: 32    move times: 84
n=17 ... ok!    exchange times: 34    move times: 132
n=18 ... ok!    exchange times: 37    move times: 168
n=19 ... ok!    exchange times: 39    move times: 230
n=20 ... ok!    exchange times: 44    move times: 196

$ input command >

```

3. 交换次数变化

第二个方案还可以再改进。

例如， $N=9$ 时，第 2 步执行后，实际上中间位置的两边对称的 4 个元素基本配对，只需交换中间的两个元素即可，如下表所示。颜色表示每次要交换的元素。

左边向右交换

右边向左交换

$N=9$

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | n+1 | n+2 | n+3 | n+4 | n+5 | n+6 | n+7 | n+8 | n+9 | | | |
|---|-----|-----|-----|-----|-----|-----|-----|-----|----|------|------|------|------|------|------|------|------|-----|--------|--------|------|
| | n-8 | n-7 | n-6 | n-5 | n-4 | n-3 | n-2 | n-1 | N | 2n-8 | 2n-7 | 2n-6 | 2n-5 | 2n-4 | 2n-3 | 2n-2 | 2n-1 | 2n | 交换开始位置 | | 交换个数 |
| | a1 | a2 | a3 | a4 | a5 | a6 | a7 | a8 | a9 | b1 | b2 | b3 | b4 | b5 | b6 | b7 | b8 | b9 | 2↔n+1 | 2n-1↔n | 1 |
| 1 | a1 | b1 | a3 | a4 | a5 | a6 | a7 | a8 | b8 | a2 | b2 | b3 | b4 | b5 | b6 | b7 | a9 | b9 | 3↔n+1 | 2n-2↔n | 2 |
| 2 | a1 | B1 | a2 | b2 | a5 | a6 | a7 | b6 | b7 | a3 | a4 | b3 | b4 | b5 | a8 | b8 | a9 | b9 | 5↔n+1 | 2n-4↔n | 4 |
| 3 | a1 | B1 | a2 | b2 | a5 | a3 | a4 | b3 | b4 | a6 | a7 | b6 | b7 | b5 | a8 | b8 | a9 | b9 | 对称交换 | | |
| 4 | a1 | B1 | a2 | b2 | a3 | a4 | b3 | b4 | a5 | b5 | a6 | a7 | b6 | b7 | a8 | b8 | a9 | b9 | | | |

每次交换的个数 $count$ (初始值为 1, $count=2*count$)

已经处理的个数 $done$ (初始值为 1, $done+=count$)

左边未处理的 a 个数 $lefta$ ($lefta=n-done-count$)

左边未匹配的 a 个数 $notmatch$ ($notmatch=lefta-count$)

如上表第 2 步, $done=4$, $count=2$, $lefta=3$, 因 $lefta>count$, 所以交换到左边的 b_6b_7 有匹配的 a , 即 $a_6a_7b_6b_7$, 所以未匹配的 a 个数为 $notmatch=1$, 即 a_5

接下来, $count=4$, 则 $notmatch<count$, 所以循环结束, 因 $notmatch>0$, 则对称交换 $count=4$ 个数据, 且要交换中间的两个数据以满足要求

然后左边的数据循环左移 $lefta$ 位, 右边的数据循环右移 $lefta$ 位

最后递归处理 $n=lefta=1$, 程序结束。

$N=7$

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | n+1 | n+2 | n+3 | n+4 | n+5 | n+6 | n+7 | | | |
|---|-----|-----|-----|-----|----------|-----|----|----------|------|------|------|------|------|-----|-------------------------|--------------------------|---|
| | n-6 | n-5 | n-4 | n-3 | n-2 | n-1 | n | 2n-6 | 2n-5 | 2n-4 | 2n-3 | 2n-2 | 2n-1 | 2n | | | |
| | a1 | a2 | a3 | a4 | a5 | a6 | a7 | b1 | b2 | b3 | b4 | b5 | b6 | b7 | $2 \leftrightarrow n+1$ | $2n-1 \leftrightarrow n$ | 1 |
| 1 | a1 | b1 | a3 | a4 | a5 | a6 | b6 | a2 | b2 | b3 | b4 | b5 | a7 | b7 | $3 \leftrightarrow n+1$ | $2n-2 \leftrightarrow n$ | 2 |
| 2 | a1 | b1 | a2 | b2 | a5 | b4 | b5 | a3 | a4 | b3 | a6 | a6 | a7 | b7 | $5 \leftrightarrow n+1$ | $2n-4 \leftrightarrow n$ | 4 |
| 3 | a1 | b1 | a2 | b2 | a5 | a3 | a4 | b4 | b5 | b3 | a6 | a6 | a7 | b7 | 对称交换 | | |
| 4 | a1 | b1 | a2 | b2 | a3 | a4 | a5 | b3 | b4 | b5 | a6 | a6 | a7 | b7 | | | |
| | | | | | 循环左移 1 位 | | | 循环右移 1 位 | | | | | | | | | |

第 2 步, $count=2$, 交换 a_3a_4 , a_6b_6 后, $lefta=1$, $notmatch=-1$, 即没有与交换到左边的 b 匹配的 a 了, 故对称交换 $count$ 个数据。

该方案需要判断 $lefta$ 与该次交换个数 $count$ 的关系:

- 若 $lefta \geq count$, 如 $n=9$, 交换到左边的 b 可以配对 (同时, 交换到右边的 a 也配对), 故只需将这些数据循环左移 $notmatch$ 位;
- 若 $lefta < count$, 如 $n=7$, 交换到左边的 b 不能配对 (同时, 交换到右边的 a 也不配对), 故要将左边不能配对的 b 和右边不能配对的 a 交换, 如上表所示, 然后再将左边没有配对的 a 循环左移 $lefta$ 位, 将右边没有配对的 b 循环右移 $lefta$ 位;

时间复杂度:

该算法的主要时间有两个: 交换和移动, 交换次数为 $T_1=O(n)$, 移动次数计算如下:

$$n=1+1+2+4+8+\dots+2^k+(2^k+c)=2^{k+1}+2^k+c$$

对 2^k+c 个元素循环移动 c 次, 共移动 $T_2=c(2^k+c)$ 次, 其中 $c<2^k$, 如果 $c=2^k$, 则不需任何移动, 交换即可完成, 即 $T_2=0$;

故 $2^{k+1}<n<2^{k+1}+2^k+2^k$, 即 $2^{k+1}<n<2^{k+2}$, $T_2<2^k * (2^k + 2^k)=2^{2k+1}<n^2/2$

故 $T=T_1+T_2=O(n^2)$

<http://blog.csdn.net/livelylittlefish>, <http://www.abo321.org>

$n=13$, $n=15$ 的例子

$N=13$

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | n+1 | n+2 | n+3 | n+4 | n+5 | n+6 | n+7 | n+8 | n+9 | n+10 | n+11 | n+12 | n+13 | | | |
|---|------|------|------|-----|-----|-----|-----|-----|----------|-----|-----|-----|-----|----------|-------|-------|------|------|------|------|------|------|------|------|------|------|-------------------------|--------------------------|---|
| | n-12 | n-11 | n-10 | n-9 | n-8 | n-7 | n-6 | n-5 | n-4 | n-3 | n-2 | n-1 | n | 2n-12 | 2n-11 | 2n-10 | 2n-9 | 2n-8 | 2n-7 | 2n-6 | 2n-5 | 2n-4 | 2n-3 | 2n-2 | 2n-1 | 2n | | | |
| | a1 | a2 | a3 | a4 | a5 | a6 | a7 | a8 | a9 | a10 | a11 | a12 | a13 | b1 | b2 | b3 | b4 | b5 | b6 | b7 | b8 | b9 | b10 | b11 | b12 | b13 | $2 \leftrightarrow n+1$ | $2n-1 \leftrightarrow n$ | 1 |
| 1 | a1 | b1 | a3 | a4 | a5 | a6 | a7 | a8 | a9 | a10 | a11 | a12 | b12 | a2 | b2 | b3 | b4 | b5 | b6 | b7 | b8 | b9 | b10 | b11 | a13 | b13 | $3 \leftrightarrow n+1$ | $2n-2 \leftrightarrow n$ | 2 |
| 2 | a1 | b1 | a2 | b2 | a5 | a6 | a7 | a8 | a9 | a10 | a11 | b10 | b11 | a3 | a4 | b3 | b4 | b5 | b6 | b7 | b8 | b9 | a12 | b12 | a13 | b13 | $5 \leftrightarrow n+1$ | $2n-4 \leftrightarrow n$ | 4 |
| 3 | a1 | b1 | a2 | b2 | a3 | b3 | a4 | b4 | a9 | b6 | b7 | b8 | b9 | a5 | a6 | a7 | a8 | b5 | a10 | b10 | a11 | b11 | a12 | b12 | a13 | b13 | | | |
| 4 | a1 | b1 | a2 | b2 | a3 | b3 | a4 | b4 | a9 | a5 | a6 | a7 | a8 | b6 | b7 | b8 | b9 | b5 | a10 | b10 | a11 | b11 | a12 | b12 | a13 | b13 | 互换 | | |
| 5 | a1 | b1 | a2 | b2 | a3 | b3 | a4 | b4 | a5 | a6 | a7 | a8 | a9 | b5 | b6 | b7 | b8 | b9 | a10 | b10 | a11 | b11 | a12 | b12 | a13 | b13 | | | |
| | | | | | | | | | 循环左移 1 位 | | | | | 循环右移 1 位 | | | | | | | | | | | | | | | |

$N=15$

| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|------|------|------|------|------|-----|-----|-----|----------|-----|-----|-----|-----|-----|-----|----------|-------|-------|-------|-------|------|------|------|------|------|------|------|------|------|-----|-------|--------|---|--|--|
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | n+1 | n+2 | n+3 | n+4 | n+5 | n+6 | n+7 | n+8 | n+9 | n+10 | n+11 | n+12 | n+13 | | | | | | | |
| n-14 | n-13 | n-12 | n-11 | n-10 | n-9 | n-8 | n-7 | n-6 | n-5 | n-4 | n-3 | n-2 | n-1 | n | 2n-14 | 2n-13 | 2n-12 | 2n-11 | 2n-10 | 2n-9 | 2n-8 | 2n-7 | 2n-6 | 2n-5 | 2n-4 | 2n-3 | 2n-2 | 2n-1 | 2n | | | | | |
| a1 | a2 | a3 | a4 | a5 | a6 | a7 | a8 | a9 | a10 | a11 | a12 | a13 | a14 | a15 | b1 | b2 | b3 | b4 | b5 | b6 | b7 | b8 | b9 | b10 | b11 | b12 | b13 | b14 | b15 | 2↔n+1 | 2n-1↔n | 1 | | |
| a1 | b1 | a3 | a4 | a5 | a6 | a7 | a8 | a9 | a10 | a11 | a12 | a13 | a14 | b14 | a2 | b2 | b3 | b4 | b5 | b6 | b7 | b8 | b9 | b10 | b11 | b12 | b13 | a15 | b15 | 3↔n+1 | 2n-2↔n | 2 | | |
| a1 | b1 | a2 | b2 | a5 | a6 | a7 | a8 | a9 | a10 | a11 | a12 | a13 | b12 | b13 | a3 | b3 | b3 | b4 | b5 | b6 | b7 | b8 | b9 | b10 | b11 | a14 | b14 | a15 | b15 | 5↔n+1 | 2n-4↔n | 4 | | |
| a1 | b1 | a2 | b2 | a3 | b3 | a4 | b4 | a9 | a10 | a11 | b8 | b9 | b10 | b11 | a5 | a6 | a7 | a8 | b5 | b6 | b7 | a12 | b12 | a13 | b13 | a14 | b14 | a15 | b15 | | | | | |
| | | | | | | | | a9 | a10 | a11 | a5 | a6 | a7 | a8 | b8 | b9 | b10 | b11 | b5 | b6 | b7 | | | | | | | | | | 互换 | | | |
| | | | | | | | | 循环左移 3 位 | | | | | | | 循环右移 3 位 | | | | | | | | | | | | | | | | | | | |

算法思想:

以 $N=9$ 为例(中间的竖表示中间位置):

a1 a2 a3 a4 a5 a6 a7 a8 a9 | b1 b2 b3 b4 b5 b6 b7 b8 b9

头尾的元素不需任何操作

1. 左边从位置 $left=2$ 开始, 右边从位置 $n+1$ 开始, 向右交换 $count=1$ 个元素, 即 a2, b1 交换
右边从位置 $right=2n-1$ 开始, 左边从位置 n 开始, 向左交换 $count=1$ 个元素, 即 b8, a9 交换
序列变为:

a1 b1 a3 a4 a5 a6 a7 a8 b8 | a2 b2 b3 b4 b5 b6 b7 a9 b9

故已经成功放好位置的有(a1, b1), (a9, b9)

其中(a8, b8), (a2, b2) 也配对, 只需将其交换到相应的位置即可

2. 左边从位置 $left=3$ 开始, 右边从位置 $n+1$ 开始, 向右交换 $count=2$ 个元素, 即 a3 a4 和 a2 b2 交换
右边从位置 $right=2n-2$ 开始, 左边从位置 n 开始, 向左交换 $count=2$ 个元素, 即 b6 b7 和 a8 b8 交换
序列变为:

a1 b1 a2 b2 a5 a6 a7 b6 b7 | a3 a4 b3 b4 b5 a8 b8 a9 b9

故又成功放好位置的有(a2, b2), (a8, b8)

其中(a6, a7, b6, b7), (a3, a4, b3, b4) 只需交换中间两个元素也配对, 只需将其交换到相应的位置即可

3. 左边从位置 $left=5$ 开始, 右边从位置 n 开始, 已不能满足交换 $count=4$ 个元素的要求, 故退出循环
从中间位置对称交换 $count=4$ 个元素

序列变为:

a1 b1 a2 b2 a5 a3 b3 a4 b4 | a6 b6 a7 b7 b5 a8 b8 a9 b9

此时只需将配对成功的(a3, b3), (a4, b4), (a6, b6), (a7, b7) 移动相应的位置即可

4. 左边从位置 $left=5$ 开始的 $size=n-left+1=5$ 个元素循环左移 1 位

右边从位置 $n+1$ 开始的 $size=5$ 个元素循环右移 1 位

序列变为:

a1 b1 a2 b2 a3 b3 a4 b4 a5 | b5 a6 b6 a7 b7 a8 b8 a9 b9

5. 至此, 将原来的序列缩小为:

a5 | b5

对该序列进行上述操作, 直到所有元素都放到正确的位置

为方便比较, 序列变化如下:

| | |
|---|---|
| <u>a1</u> <u>a2</u> <u>a3</u> <u>a4</u> <u>a5</u> <u>a6</u> <u>a7</u> <u>a8</u> <u>a9</u> | <u>b1</u> <u>b2</u> <u>b3</u> <u>b4</u> <u>b5</u> <u>b6</u> <u>b7</u> <u>b8</u> <u>b9</u> |
| <u>a1</u> <u>b1</u> <u>a3</u> <u>a4</u> <u>a5</u> <u>a6</u> <u>a7</u> <u>a8</u> <u>b8</u> | <u>a2</u> <u>b2</u> <u>b3</u> <u>b4</u> <u>b5</u> <u>b6</u> <u>b7</u> <u>a9</u> <u>b9</u> |
| <u>a1</u> <u>b1</u> <u>a2</u> <u>b2</u> <u>a5</u> <u>a6</u> <u>a7</u> <u>b6</u> <u>b7</u> | <u>a3</u> <u>a4</u> <u>b3</u> <u>b4</u> <u>b5</u> <u>a8</u> <u>b8</u> <u>a9</u> <u>b9</u> |
| <u>a1</u> <u>b1</u> <u>a2</u> <u>b2</u> <u>a5</u> <u>a3</u> <u>b3</u> <u>a4</u> <u>b4</u> | <u>a6</u> <u>b6</u> <u>a7</u> <u>b7</u> <u>b5</u> <u>a8</u> <u>b8</u> <u>a9</u> <u>b9</u> |
| <u>a1</u> <u>b1</u> <u>a2</u> <u>b2</u> <u>a3</u> <u>b3</u> <u>a4</u> <u>b4</u> <u>a5</u> | <u>b5</u> <u>a6</u> <u>b6</u> <u>a7</u> <u>b7</u> <u>a8</u> <u>b8</u> <u>a9</u> <u>b9</u> |

源程序如下:

```

/*****
 * 输入a1,a2,...,an,b1,b2,...,bn
 * 在O(n)的时间,O(1)的空间
 * 将这个序列顺序改为a1,b1,a2,b2,a3,b3,...,an,bn
 *****/

```

```

#include <stdio.h>
#include <string.h>
#include <CONIO.H>

#define MAXSIZE 2*1000

```

```

int exchangetimes=0; //交换次数
int movetimes=0;    //移动次数

```

//交换两个数据

```
void swap(int *x,int *y)
```

```

{
    int t;
    t=*x;
    *x=*y;
    *y=t;

```

```
    exchangetimes++;
```

```
}
```

//将数组中的元素循环左移x位,n为数组的元素个数

```

void rotate_left(int a[],int n,int x)
{
    int t;
    for(int i=0;i<x;i++)
    {
        t=a[0];
        for(int j=0;j<n-1;j++)
            a[j]=a[j+1];
        a[n-1]=t;
    }

    movetimes+=n*x;
}

```

//将数组中的元素循环右移x位,n为数组的元素个数

```

void rotate_right(int a[],int n,int x)
{
    int t;
    for(int i=0;i<x;i++)
    {
        t=a[n-1];
        for(int j=n-1;j>0;j--)
            a[j]=a[j-1];
        a[0]=t;
    }

    movetimes+=n*x;
}

```

//按要求交换序列(假设元素从下标1开始存放)

```

void exchange(int a[],int m)
{
    int n=m/2;

    if(n==1)        //a1,b1 ==> 不需交换
        return;
    else if(n==2)    //a1,a2,b1,b2 ==> 只需交换中间的两个数据
    {
        swap(a+1,a+2);
        return;
    }

    int done;        //已经处理的数据个数
    int left;         //左边开始交换的位置
    int right;        //右边开始交换的位置
    int count;        //每次交换的数据个数
    int lefta;         //左边未处理的a个数
    int notmatch;     //左边未匹配的a个数

    //初始化
    done=1;
    left=1;           //左边从位置1开始向右交换
    right=2*n-2;      //右边从位置2*n-2开始向左交换
    count=1;          //每次交换count个数据
    lefta=notmatch=n-1;

    do
    {
        //交换连续的count个元素
        for(int j=0;j<count;j++)
            swap(a+left+j,a+n+j);

        //若notmatch=0,该交换不能进行,否则为重复交换
        if(notmatch>0)
        {
            for(j=0;j<count;j++)
                swap(a+right-j,a+n-1-j);
        }
        else
            break;
    }
}

```

```

//交换后将其调整为要求的序列
if(count>=4)
{
    exchange(a+left,count);
    exchange(a+right-count+1,count);
}

//重新调整各变量
done+=count;
lefta=n-done-count;
notmatch=lefta-count;

left=left+count;
right=right-count;

count*=2;

if(notmatch<count)
    break;
}while(1);

if(notmatch<0)
{
    count/=2;

    //由中间对称交换count个数据
    for(int j=0;j<count;j++)
        swap(a+n-count+j,a+n+j);

    int size=n-left;
    //左边的数据循环左移lefta位
    rotate_left(a+left,size,lefta);

    //右边的数据循环右移lefta位
    rotate_right(a+n,size,lefta);

    int newm=2*size;
    //递归调用
    exchange(a+left,newm);
}
Else
{
    //由中间对称交换count个数据
    for(int j=0;j<count;j++)
        swap(a+n-count+j,a+n+j);

    //递归调用中间对称的count个数据
    exchange(a+n-count,count);
    exchange(a+n,count);

    if(notmatch>0)
    {
        int size=n-left;
        //左边的数据循环左移notmatch位
        rotate_left(a+left,size,notmatch);

        //右边的数据循环右移notmatch位
        rotate_right(a+n,size,notmatch);

        int newm=2*notmatch;
        //递归调用
        exchange(a+left+count,newm);
    }
}

//显示菜单
void show_menu()
{
    printf("-----\n");
    printf("input command to test the program\n");
    printf("    i or I : input n to test\n");
}

```

```

printf("    t or T : test program\n");
printf("    q or Q : quit\n");
printf("-----\n");
printf("$ input command >");
}

//显示数据
void display(int a[],int n)
{
    for(int i=0; i<n; i++)
        printf("%3d",a[i]);
    printf("\n");
}

//检查交换是否正确
bool check(int a[],int n)
{
    int i;

    for(i=0; i<n-2; i+=2)
    {
        if(a[i]!=i/2+1)
            return false;
    }

    for(i=1; i<n-1; i+=2)
    {
        if(a[i]!=(n+i+1)/2)
            return false;
    }

    return true;
}

void main()
{
    int a[MAXSIZE];
    int n;
    char sinput[10];

    show_menu();

    scanf("%s",sinput);
    while(strcmp(sinput,"q")!=0)
    {
        if(strcmp(sinput,"i")==0)
        {
            printf("    please input n:");
            scanf("%d",&n);

            //假设数组的第0个元素为0,且不对其进行操作
            //且假设数组中的数据为1,2,3,...,n,n+1,n+2,...,2n
            for(int i=0; i<2*n; i++) //初始化
                a[i]=i+1;

            display(a,2*n);

            exchangetimes=0;
            movetimes=0;

            //交换
            exchange(a,2*n);
            display(a,2*n);

            printf("\n    exchange times: %d\n",exchangetimes);
            printf("    move times: %d\n\n",movetimes);
        }
        else if(strcmp(sinput,"t")==0)
        {
            int n1,n2;
            printf("    please input the begin number:");
            scanf("%d",&n1);

```

```

printf(" please input the end number:");
scanf("%d",&n2);

printf(" press any key to start ...\n\n");
getch();

for(int i=n1;i<=n2;i++)
{
    //初始化
    for(int j=0; j<2*i;j++)
        a[j]=j+1;

    exchangetimes=0;
    movetimes=0;
    exchange(a,2*i);

    if(check(a,2*i))
        //printf(" n=%d ... ok!\n",i);
        printf(" n=%d ... ok!      exchange times: %d      move times: %d\n",i,exchangetimes,movetimes);
    else
        printf(" n=%d ... wrong!\n",i);
}

printf("\n");
}

//输入命令
printf("$ input command >");
scanf("%s",sinput);
}
}

```

运行结果如下:

```

C:\Documents and Settings\yuzubo\桌面\test\Debug\exchange2.exe
-----
input command to test the program
i or I : input n to test
t or T : test program
q or Q : quit
-----
$ input command >i
please input n:7
1 2 3 4 5 6 7 8 9 10 11 12 13 14
1 8 2 9 3 10 4 11 5 12 6 13 7 14

exchange times: 11
move times: 6

$ input command >i
please input n:9
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18
1 10 2 11 3 12 4 13 5 14 6 15 7 16 8 17 9 18

exchange times: 12
move times: 10

$ input command >

```

```

C:\Documents and Settings\yuzubo\桌面\test\Debug\exchange2.exe
-----
input command to test the program
i or I : input n to test
t or T : test program
q or Q : quit
-----
$ input command >t
please input the begin number:11
please input the end number:20
press any key to start ...

n=11 ... ok!   exchange times: 15   move times: 42
n=12 ... ok!   exchange times: 24   move times: 0
n=13 ... ok!   exchange times: 24   move times: 16
n=14 ... ok!   exchange times: 29   move times: 24
n=15 ... ok!   exchange times: 31   move times: 48
n=16 ... ok!   exchange times: 32   move times: 0
n=17 ... ok!   exchange times: 32   move times: 18
n=18 ... ok!   exchange times: 33   move times: 40
n=19 ... ok!   exchange times: 35   move times: 66
n=20 ... ok!   exchange times: 36   move times: 96

$ input command >

```

很明显，该方案比第二种方案效率高。

4. 不需移动，通过交换完成

对上述第三个方案再改进。

例如，N=9 时，第 2 步执行后，实际上中间位置的两边对称的 4 个元素基本配对，只需交换中间的两个元素即可，如下表所示。颜色表示每次要交换的元素。

左边向右交换

右边向左交换

N=9

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | n+1 | n+2 | n+3 | n+4 | n+5 | n+6 | n+7 | n+8 | n+9 | | | |
|---|-----|-----|-----|-----|-----|---------------|-----|-----|----|------|---------------|------|------|------|------|------|------|-----|--------|--------|------|
| | n-8 | n-7 | n-6 | n-5 | n-4 | n-3 | n-2 | n-1 | N | 2n-8 | 2n-7 | 2n-6 | 2n-5 | 2n-4 | 2n-3 | 2n-2 | 2n-1 | 2n | 交换开始位置 | | 交换个数 |
| | a1 | a2 | a3 | a4 | a5 | a6 | a7 | a8 | a9 | b1 | b2 | b3 | b4 | b5 | b6 | b7 | b8 | b9 | 2↔n+1 | 2n-1↔n | 1 |
| 1 | a1 | b1 | a3 | a4 | a5 | a6 | a7 | a8 | b8 | a2 | b2 | b3 | b4 | b5 | b6 | b7 | a9 | b9 | 3↔n+1 | 2n-2↔n | 2 |
| 2 | a1 | B1 | a2 | b2 | a5 | a6 | a7 | b6 | b7 | a3 | a4 | b3 | b4 | b5 | a8 | b8 | a9 | b9 | 5↔n+1 | 2n-4↔n | 4 |
| 3 | a1 | B1 | a2 | b2 | X1 | Y1=(a6 a7 b7) | | | X2 | X3 | Y2=(a4 b3 b4) | | | X4 | a8 | b8 | a9 | b9 | 对称交换 | | |
| | a1 | B1 | a2 | b2 | X3 | Y2 | | | X4 | X1 | Y1 | | | X2 | a8 | b8 | a9 | b9 | | | |
| | a1 | B1 | a2 | b2 | X3 | Y2 | | | X1 | X4 | Y1 | | | X2 | a8 | b8 | a9 | b9 | | | |
| 4 | a1 | B1 | a2 | b2 | A3 | A4 | B3 | B4 | A5 | B5 | A6 | A7 | B6 | B7 | a8 | b8 | a9 | b9 | | | |
| 5 | a1 | B1 | a2 | b2 | A3 | B3 | A4 | B4 | A5 | B5 | A6 | B6 | A7 | B7 | a8 | b8 | a9 | b9 | | | |

交换x1,x3; 交换x2,x4; 再交换中间的x1,x4; 交换y1,y2;

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | n+1 | n+2 | n+3 | n+4 | n+5 | n+6 | n+7 | n+8 | n+9 | | | |
|---|-----|-----|-----|-----|-----|---------------|-----|-----|----|------|---------------|------|------|------|------|------|------|-----|--------|--------|------|
| | n-8 | n-7 | n-6 | n-5 | n-4 | n-3 | n-2 | n-1 | n | 2n-8 | 2n-7 | 2n-6 | 2n-5 | 2n-4 | 2n-3 | 2n-2 | 2n-1 | 2n | 交换开始位置 | | 交换个数 |
| | a1 | a2 | a3 | a4 | a5 | a6 | a7 | a8 | a9 | b1 | b2 | b3 | b4 | b5 | b6 | b7 | b8 | b9 | 2↔n+1 | 2n-1↔n | 1 |
| 1 | a1 | b1 | a3 | a4 | a5 | a6 | a7 | a8 | b8 | a2 | b2 | b3 | b4 | b5 | b6 | b7 | a9 | b9 | 3↔n+1 | 2n-2↔n | 2 |
| 2 | a1 | B1 | a2 | b2 | a5 | a6 | a7 | b6 | b7 | a3 | a4 | b3 | b4 | b5 | a8 | b8 | a9 | b9 | 5↔n+1 | 2n-4↔n | 4 |
| 3 | a1 | B1 | a2 | b2 | X1 | Y1=(a6 a7 b7) | | | X2 | X3 | Y2=(a4 b3 b4) | | | X4 | a8 | b8 | a9 | b9 | 对称交换 | | |
| | a1 | B1 | a2 | b2 | X3 | Y2 | | | X4 | X1 | Y1 | | | X2 | a8 | b8 | a9 | b9 | | | |
| | a1 | B1 | a2 | b2 | X3 | Y2 | | | X1 | X4 | Y1 | | | X2 | a8 | b8 | a9 | b9 | | | |
| 4 | a1 | B1 | a2 | b2 | A3 | A4 | B3 | B4 | A5 | B5 | A6 | A7 | B6 | B7 | a8 | b8 | a9 | b9 | | | |
| 5 | a1 | B1 | a2 | b2 | A3 | B3 | A4 | B4 | A5 | B5 | A6 | B6 | A7 | B7 | a8 | b8 | a9 | b9 | | | |

时间复杂度:

该算法的主要时间有两个: 交换和移动, 交换次数为 $T_1=O(n)$, 移动次数计算如下:

$$n=1+1+2+4+8+\dots+2^k+(2^k+c)=2^{k+1}+2^k+c$$

对 2^k+c 个元素循环移动 c 次, 共移动 $T_2=c(2^k+c)$ 次, 其中 $c<2^k$, 如果 $c=2^k$, 则不需任何移动, 交换即可完成, 即 $T_2=0$;

$$\text{故 } 2^{k+1}<n<2^{k+1}+2^k+2^k, \text{ 即 } 2^{k+1}<n<2^{k+2}, T_2<2^k * (2^k + 2^k)=2^{2k+1}<n^2/2$$

$$\text{故 } T=T_1+T_2=O(n^2)$$

算法思想:

以 $N=9$ 为例(中间的竖表示中间位置):

a1 a2 a3 a4 a5 a6 a7 a8 a9 | b1 b2 b3 b4 b5 b6 b7 b8 b9

头尾的元素不需任何操作

1. 左边从位置 $left=2$ 开始, 右边从位置 $n+1$ 开始, 向右交换 $count=1$ 个元素, 即 a2, b1 交换
右边从位置 $right=2n-1$ 开始, 左边从位置 n 开始, 向左交换 $count=1$ 个元素, 即 b8, a9 交换
序列变为:

a1 b1 a3 a4 a5 a6 a7 a8 b8 | a2 b2 b3 b4 b5 b6 b7 a9 b9

故已经成功放好位置的有(a1, b1), (a9, b9)

其中(a8, b8), (a2, b2) 也配对, 只需将其交换到相应的位置即可

2. 左边从位置 $left=3$ 开始, 右边从位置 $n+1$ 开始, 向右交换 $count=2$ 个元素, 即 a3 a4 和 a2 b2 交换
右边从位置 $right=2n-2$ 开始, 左边从位置 n 开始, 向左交换 $count=2$ 个元素, 即 b6 b7 和 a8 b8 交换
序列变为:

a1 b1 a2 b2 a5 a6 a7 b6 b7 | a3 a4 b3 b4 b5 a8 b8 a9 b9

故又成功放好位置的有(a2, b2), (a8, b9)

3. 左边从位置 $left=5$ 开始, 右边从位置 n 开始, 已不能满足交换 $count=4$ 个元素的要求, 故退出循环

4. 序列缩小为a5 a6 a7 b6 b7 | a3 a4 b3 b4 b5, 对序列中没有放好的数据按快处理

将序列看作: $x1=(a5)$ $y1=(a6\ a7\ b6)$ $x2=(b7)$ | $x3=(a3)$ $y2=(a4\ b3\ b4)$ $x4=(b5)$

交换 $x1, x3$; 交换 $x2, x4$; 再交换中间的 $x1, x4$; 交换 $y1, y2$;

此时序列变为S1: a3 a4 b3 b4 a5 b5 a6 a7 b6 b7

5. 若交换到左边的 b 有配对的 a, 则中间的序列 $S2=(a5\ b6)$ 作为新的序列; 否则将 S1 作为新的序列; 对该序列进行上述操作, 直到所有元素都放到正确的位置

此例中, (a6, a7, b6, b7), (a3, a4, b3, b4) 以配对, 只需交换中间的两个元素即可, 序列缩小 a5 b5

为方便比较, 序列变化如下:

| | | |
|---|--|---|
| <u>a1</u> a2 a3 a4 a5 a6 a7 a8 a9 | | b1 b2 b3 b4 b5 b6 b7 b8 b9 |
| <u>a1</u> <u>b1</u> a3 a4 a5 a6 a7 <u>a8</u> <u>b8</u> | | <u>a2</u> <u>b2</u> b3 b4 b5 b6 b7 <u>a9</u> <u>b9</u> |
| <u>a1</u> <u>b1</u> <u>a2</u> <u>b2</u> a5 <u>a6</u> a7 <u>b6</u> <u>b7</u> | | <u>a3</u> <u>a4</u> <u>b3</u> <u>b4</u> b5 <u>a8</u> <u>b8</u> <u>a9</u> <u>b9</u> |
| <u>a1</u> <u>b1</u> <u>a2</u> <u>b2</u> <u>a3</u> <u>a4</u> <u>b3</u> <u>b4</u> <u>a5</u> | | <u>b5</u> <u>a6</u> <u>b6</u> <u>a7</u> <u>b7</u> <u>a8</u> <u>b8</u> <u>a9</u> <u>b9</u> |
| <u>a1</u> <u>b1</u> <u>a2</u> <u>b2</u> <u>a3</u> <u>b3</u> <u>a4</u> <u>b4</u> <u>a5</u> | | <u>b5</u> <u>a6</u> <u>b6</u> <u>a7</u> <u>b7</u> <u>a8</u> <u>b8</u> <u>a9</u> <u>b9</u> |

源程序如下:

```

/*****
 * 输入a1,a2,...,an,b1,b2,...,bn
 * 在O(n)的时间,O(1)的空间
 * 将这个序列顺序改为a1,b1,a2,b2,a3,b3,...,an,bn
 *****/

```

```

#include <stdio.h>
#include <string.h>
#include <CONIO.H>

```

```

#define MAXSIZE 2*100000

```

```

int exchangetimes=0; //交换次数

```

```

//交换两个数据

```

```

void swap(int *x,int *y)

```

```

{
    int t;
    t=*x;
    *x=*y;
    *y=t;

```

```

    exchangetimes++;

```

```

}

```

```

//按要求交换序列(假设元素从下标1开始存放)

```

```

void exchange(int a[],int m)

```

```

{
    int n=m/2;

```

```

    if(n==1) //a1,b1 ==>不需交换

```

```

    return;
else if(n==2)    //a1,a2,b1,b2 ==>只需交换中间的两个数据
{
    swap(a+1,a+2);
    return;
}

int done;        //已经处理的数据个数
int left;        //左边开始交换的位置
int right;       //右边开始交换的位置
int count;       //每次交换的数据个数
int lefta;       //左边未处理的a个数
int notmatch;    //左边未匹配的a个数

//初始化
done=1;
left=1;          //左边从位置1开始向右交换
right=2*n-2;     //右边从位置2*n-2开始向左交换
count=1;         //每次交换count个数据
lefta=notmatch=n-1;

while(1)
{
    //左边从left开始和右边从n开始向右交换count个数据
    for(int j=0;j<count;j++)
        swap(a+left+j,a+n+j);

    //右边从right开始和左边从n-1开始向左交换count个数据
    for(j=0;j<count;j++)
        swap(a+right-j,a+n-1-j);

    //交换后将其调整为要求的序列
    if(count>=4)
    {
        exchange(a+left,count);
        exchange(a+right-count+1,count);
    }

    //重新调整各变量
    done+=count;
    lefta=n-done-count;
    notmatch=lefta-count;
    left=left+count;
    right=right-count;
    count*=2;

    if(notmatch<count)
        break;
}

int x,y;

//左边剩下的a不能与从右边交换过来的b配对
//如n=13时,上面的循环结束后变为: a9 (b6 b7 b8) b9 | a5 (a6 a7 a8) b5
//此时,lefta=1,notmatch<0,分块交换,各个块如下
//x1=(a9), y1=(b6 b7 b8), x2=(b9), x3=(a5), y2=(a6 a7 a8), x4=(b5)
//上述序列变为 x1 y1 x2 x3 y2 x4, x的长度均为1,y的长度均为3
//x1 x3交换,x2 x4交换==>x3 x4 x1 x2,然后中间的x4 x1交换==>x3 x1 x4 x2
//y1 y2交换==>y2 y1
//上述6块经4次交换后变为 x3 y2 x1 x4 y1 x2
//n=13时,经上述交换后变为 a5 a6 a7 a8 a9 b5 b6 b7 b8 b9
if(notmatch<0)
{
    count/=2; //if n=13, then here count=4
    x=lefta;  //x块的长度,if n=13, then here x=1
}
else
{
    //递归调用中间对称的count个数据
    exchange(a+n-count,count);
}

```



```

    exchange(a+n,count);

    x=notmatch;
}

////////////////////////////////////
y=count-x; //y块的长度,if n=13, then heare y=3
//左边从left开始和右边从n开始向右交换x个数据,即x1 x3交换
//右边从right开始和左边从n-1开始向左交换x个数据,即x2 x4交换
for(int j=0;j<x;j++)
{
    swap(a+left+j,a+n+j);    //左边向左交换
    swap(a+right-j,a+n-1-j); //右边向左交换
}

//交换到中间的x1 x4交换
for(j=0;j<x;j++)
    swap(a+n-x+j,a+n+j);

//交换y1 y2数据块
for(j=0;j<y;j++)
    swap(a+left+x+j,a+n+x+j);
////////////////////////////////////

//处理余下的序列
if(notmatch<0)
{
    int newm=2*(n-left); //或者size=x+count
    exchange(a+left,newm);
}
else if(notmatch>0)
{
    int newm=2*notmatch;
    exchange(a+left+count,newm);
}
}

//显示菜单
void show_menu()
{
    printf("-----\n");
    printf("input command to test the program\n");
    printf("    i or I : input n to test\n");
    printf("    t or T : test program\n");
    printf("    q or Q : quit\n");
    printf("-----\n");
    printf("$ input command >");
}

//显示数据
void display(int a[],int n)
{
    for(int i=0; i<n; i++)
        printf("%3d",a[i]);
    printf("\n");
}

//检查交换是否正确
bool check(int a[],int n)
{
    int i;

    for(i=0;i<n-2;i+=2)
    {
        if(a[i]!=i/2+1)
            return false;
    }

    for(i=1;i<n-1;i+=2)
    {
        if(a[i]!=(n+i+1)/2)
            return false;
    }
}

```

```

    }

    return true;
}

void main()
{
    int a[MAXSIZE];
    int n;
    char sinput[10];

    show_menu();

    scanf("%s",sinput);
    while(stricmp(sinput,"q")!=0)
    {
        if(stricmp(sinput,"i")==0)
        {
            printf("  please input n:");
            scanf("%d",&n);

            //且假设数组中的数据为1,2,3,...,n,n+1,n+2,...,2n
            for(int i=0; i<2*n;i++) //初始化
                a[i]=i+1;

            display(a,2*n);

            exchangetimes=0;

            //交换
            exchange(a,2*n);
            display(a,2*n);

            printf("\n  exchange times: %d\n",exchangetimes);
        }
        else if(stricmp(sinput,"t")==0)
        {
            int n1,n2;
            printf("  please input the begin number:");
            scanf("%d",&n1);
            printf("  please input the  end  number:");
            scanf("%d",&n2);

            printf("  press any key to start ..\n\n");
            getch();

            for(int i=n1;i<=n2;i++)
            {
                //初始化
                for(int j=0;j<2*i;j++)
                    a[j]=j+1;

                exchangetimes=0;
                exchange(a,2*i);

                if(check(a,2*i))
                    printf("  n=%d ... ok!    exchange times: %d\n",i,exchangetimes);
                else
                    printf("  n=%d ... wrong!\n",i);
            }

            printf("\n");
        }

        //输入命令
        printf("$ input command >");
        scanf("%s",sinput);
    }
}

```

运行结果如下：

```
C:\Documents and Settings\yuzubo\桌面\test\Debug\exchange3.exe

input command to test the program
i or I : input n to test
t or T : test program
q or Q : quit

$ input command >i
please input n:7
1 2 3 4 5 6 7 8 9 10 11 12 13 14
1 8 2 9 3 10 4 11 5 12 6 13 7 14

exchange times: 13

$ input command >i
please input n:9
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18
1 10 2 11 3 12 4 13 5 14 6 15 7 16 8 17 9 18

exchange times: 14

$ input command >
```

```
C:\Documents and Settings\yuzubo\桌面\test\Debug\exchange3.exe

input command to test the program
i or I : input n to test
t or T : test program
q or Q : quit

$ input command >t
please input the begin number:11
please input the end number:20
press any key to start ...

n=11 ... ok! exchange times: 21
n=12 ... ok! exchange times: 24
n=13 ... ok! exchange times: 28
n=14 ... ok! exchange times: 33
n=15 ... ok! exchange times: 39
n=16 ... ok! exchange times: 32
n=17 ... ok! exchange times: 34
n=18 ... ok! exchange times: 37
n=19 ... ok! exchange times: 41
n=20 ... ok! exchange times: 44

$ input command >_
```