АНАЛИЗ
НА РЕШЕНИЕТО НА ЗАДАЧА
РАЗМЕСТВАНИЯ

В цикъл от m=(1 до N) наместваме подред всяко m=1,2, .. да дойде на мястото си.
Първо намираме на коя позиция е m. Нека числото m е на позиция j. Понеже не е на мястото си, явно j>m. Ако предното число не е j, то разменяме местата на m и предното число. Защо трябва a[j-1] ≠ j? Това е така защото:
P= 1 2 6 5 3 4 Стигнали сме до m=3.То е на позиция j=5. Ако разменим 5 и 3, ще се получи:
P= 1 2 6 3 **5** 4 – сега 3 се премести напред, но числото 5 се оказа на позиция 5 и повече не може да се разменя с никое, т.е. вече няма начин числото 4, което е след 5, да се придвижи наляво.
Ако беше
P= 1 2 5 6 3 4 и m=3, то разменяме 6 и 3 и се получава
P= 1 2 5 3 6 4. След това преместване, веднага променяме j да стане 4.
По този алгоритъм ако продължим, то може да разменим сега 5 и 3 защото
a[j-1] = 5 ≠ j = 4.
Става P= 1 2 3 5 6 4 и j=3.
Понеже j=m, т.е. 3 си отиде на позиция 3, цикъла се завърта и m=4.
Нека сега P= 1 2 7 5 6 3 4, m=3, j=6 и a[j-1]=j, т.е. не може да разменим 6 и 3. Връщаме се от j-1 назад и търсим първото k за което a[k] ≠ k+1. Това е при k=3, a[3] = 7 ≠ 3+1=4.
Целта е да „издърпаме" 5 и 6 наляво за да не отидат на местата си, а през това време 3 да отиде към място 3. Затова разместваме **отляво-надясно**, започвайки от i=3 до i=j-1 и разменяме a[i] и a[i+1]. Ще се получи 1 2 5 6 3 7 4. С една дума изместихме 5,6 и 3 наляво, а 7 го пратихме на мястото на 3:
1 2 7 5 6 3 4
1 2 5 6 3 7 4
Това значи,че намаляваме j, защото 3 се измести с една позиция наляво.
Обаче при P= 1 2 4 5 6 3 8 7 е по-добре да разменяме отдясно-наляво, започвайки от 3 и стигайки до 4. Ще се получи
P= 1 2 4 5 6 3 8 7
P= 1 2 3 4 5 6 8 7
Да обърнем внимание какво стойности може да има k спрямо m.
Явно до m-1 числата са подредени.
Ето всички възможни случаи:

k=4. m=3, **k>m**

индекс	1	2	3	4	5	6	7	8	9	10	11	12
P1	1	2	10	12	6	7	8	3	5	9	4	11
Отместване												
P1	1	2	10	12	6	7	8	3	10	9	12	11
P2	1	2	10	6	7	8	3	12	5	9	4	11

k=4, m=4, **k=m**

индекс	1	2	3	4	5	6	7	8	9	10	11	12
P1	1	2	3	11	6	7	8	9	4	12	10	5
Отместване												
P2	1	2	3	11	6	7	8	9	4	12	10	5
P3	1	2	3	6	7	8	9	4	11	12	10	5

k=0, m=1, **k<m**

индекс	1	2	3	4	5	6	7	8	9	10	11	12
P1	2	3	4	5	1	8	6	9	7	11	12	10
Отместване												
P2	2	3	4	5	1	8	6	9	7	11	12	10
P3	1	2	3	4	5	8	6	9	7	11	12	10

k=4. m=5, **k<m**

индекс	1	2	3	4	5	6	7	8	9	10	11	12
P1	1	2	3	4	6	7	8	5	10	12	9	11
Отместване												
P1	1	2	3	4	6	7	8	5	10	9	12	11
P2	1	2	3	4	5	6	7	8	10	12	9	11

Вижда се, че когато **k<m**, тогава "пречещите" числа /в жълто/ се изместват надясно. Ясно е, че k може да бъде или 0 или m-1, и на практика числото m си отива на място m. Изместването става с цикъла

for i := j - 1 do m swap(i, i + 1)
и правим j := m
1

Когато **k ≥ m**, тогава „пречещите" числа и m се изместват наляво, а числото, което е било на позиция k – отива на позиция m. Цикълът е:

for i := k do j - 1 swap(i, i + 1);
и правим j :=j - 1
2

В процедурата ***swap*** си разменят местата a[i] и a[j].

Псевдокод на програмата:

```
for m:=1 do n
        if a[m]=m continue
        намираме j= позицията на числото m
        while j>m
                if a[j - 1] < > j then
                        swap(j - 1, j)
                        j := j-1
                else
                        k := j - 1
                        while (k >= m) and (a[k] = k + 1)
                                k := k-1
                        if k < m
                                изместваме „пречещите" числа надясно, j:=m      1
                                else
                                изместваме „пречещите" числа наляво, j:=j-1     2
```

Източник: *Всероссийская командная олимпиада школьников по программированию*

Анализ и реализация: *Павел Петров*