

# Перебор подмножеств и перестановок

Хадиев Р.М.

# Перебор подмножеств в

Для  $n=4 - \{x_1, x_2, x_3, x_4\}$

- // (0000) -> { }
- // (0001) -> {  $x_4$  }
- // (0010) -> {  $x_3$  }
- // (0011) -> {  $x_3$   $x_4$  }
- // (0100) -> {  $x_2$  }
- // (0101) -> {  $x_2$   $x_4$  }
- // (0110) -> {  $x_2$   $x_3$  }
- // (0111) -> {  $x_2$   $x_3$   $x_4$  }
- // (1000) -> {  $x_1$  }
- // (1001) -> {  $x_1$   $x_4$  }
- // (1010) -> {  $x_1$   $x_3$  }
- // (1011) -> {  $x_1$   $x_3$   $x_4$  }
- // (1100) -> {  $x_1$   $x_2$  }
- // (1101) -> {  $x_1$   $x_2$   $x_4$  }
- // (1110) -> {  $x_1$   $x_2$   $x_3$  }
- // (1111) -> {  $x_1$   $x_2$   $x_3$   $x_4$  }

```

#include <iostream>
using namespace std;
Main() {
    Int p[100]={0}, i ,n, k;
    cin >> n;
    do {
        // печать множества
        cout << '(';
        for (i=0; i<n; i++)
            cout << p[i] << ' ';
        cout << ' ) -> {';
        for (i=1; i<=n; i++)
            if (p[i-1]) cout <<'X' << i << ' ';
        cout << '}\n';
    }
}

```

```

// переход к след. подмнож-
ву
j=n-1;
while (p[j] && j>0) {
    p[j]=0;
    j--;
}
if (j) \\ НОВЫЙ ЭЛЕМЕНТ
    p[j]=1;
}
while (j>0);
}

```

# Задача о ранце

- Задано множество товаров с весами –  
 $v_1, v_2, v_3 \dots$

Максимальная возможная загрузка ранца  $T$ .

- Описание переменных

***var***  
*x, {массив индексов для перебора подмножеств}*  
*max, {массив для максимума}*  
*v:array[0..10] of integer; {массив весов}*  
*max\_v, {максимальный вес найденной загрузки}*  
*n, j, i, t: integer; {t – предел ранца}*

- **procedure** print;
- **var** s,i:integer;
- **begin**
- write('(' ');
- **for** i:=1 **to** n **do** write(x[i], ' ');
- s:=0;
- write(') <-> { ');
- **for** i:=1 **to** n **do begin**
- **if** x[i]=1 **then** write('X[' ,i,'](' ,v[i],') ');
- s+=v[i]\*x[i];
- **end;**
- **if** s<=t **then begin**
- writeln('}= ,s, ' +');
- **if** s>max\_v **then begin // фиксация максимального**
- **подмножества**
- max\_v:=s;
- max:=x
- **end**
- **end**
- **else** writeln('} – недопустимая загрузка')
- **end;**

# Процедура печати задачи о ранце

- **begin**
- read(n,t);
- **for i:=1 to n do begin**
- read(v[i]); // вес i-го товара
- x[i]:=0 // первое множество пустое
- **end;**
- max:=x; max\_v:=0; // параметры пустого множества
- **repeat**
- print;
- j:=n;
- **while** (x[j]=1) and (j>0) **do begin**
- x[j]:=0;
- j-=1
- **end;**
- **if** j>0 **then begin** x[j]:=1
- **until** j=0;
- // печать варианта максимальной загрузки
- writeln('MAX');
- x:=max;
- print
- **end.**

# Основной модуль задачи о ранце

# $2^N \Leftrightarrow$ время работы в сутках

$$2^5=32 \Leftrightarrow 1.7e-13$$

$$2^{10}=1024 \Leftrightarrow 2.4e-10$$

$$2^{15}=32768 \Leftrightarrow 1e-8$$

$$2^{20}=1048576 \Leftrightarrow 4.9e-7$$

$$2^{25}=33554432 \Leftrightarrow 2e-5$$

$$2^{30}=1e9 \Leftrightarrow 7.5e-4 \text{ – секунда!}$$

$$2^{35}=34e9 \Leftrightarrow 0.028$$

$$2^{40}=101e10 \Leftrightarrow 1.02 \text{ – день!}$$

$$2^{45}=35e12 \Leftrightarrow 36.8$$

$$2^{50}=1.1e15 \Leftrightarrow 1309$$

- // (1,2,3,4)  $\Leftrightarrow$  (X1,X2,X3,X4)
- // (1,2,4,3)  $\Leftrightarrow$  (X1,X2,X4,X3)
- // (1,3,2,4)  $\Leftrightarrow$  (X1,X3,X2,X4)
- // (1,3,4,2)  $\Leftrightarrow$  (X1,X3,X4,X2)
- // (1,4,2,3)  $\Leftrightarrow$  (X1,X4,X2,X3)
- // (1,4,3,2)  $\Leftrightarrow$  (X1,X4,X3,X2)
- // (2,1,3,4)  $\Leftrightarrow$  (X2,X1,X3,X4)
- // (2,1,4,3)  $\Leftrightarrow$  (X1,X2,X3,X4)
- // (2,3,1,4)  $\Leftrightarrow$  (X2,X3,X1,X4)
- // (2,3,4,1)  $\Leftrightarrow$  (X2,X3,X4,X1)
- // (2,4,1,3)  $\Leftrightarrow$  (X2,X4,X1,X3)
- // (2,4,3,1)  $\Leftrightarrow$  (X2,X4,X3,X1)
- // (3,1,2,4)  $\Leftrightarrow$  (X3,X1,X2,X4)
- // (3,1,4,2)  $\Leftrightarrow$  (X3,X1,X4,X2)
- // (3,2,1,4)  $\Leftrightarrow$  (X3,X2,X1,X4)

# Перебор

## перестановок

Для n=4 – (1,2,3,4)  $\Leftrightarrow$  (X1,X2,X3,X4)

- // (3,2,4,1)  $\Leftrightarrow$  (X3,X2,X4,X1)
- // (3,4,1,2)  $\Leftrightarrow$  (X3,X4,X1,X2)
- // (3,4,2,1)  $\Leftrightarrow$  (X3,X4,X2,X1)
- // (4,1,2,3)  $\Leftrightarrow$  (X4,X1,X2,X3)
- // (4,1,3,2)  $\Leftrightarrow$  (X4,X1,X3,X2)
- // (4,2,1,3)  $\Leftrightarrow$  (X4,X2,X1,X3)
- // (4,2,3,1)  $\Leftrightarrow$  (X4,X2,X3,X1)
- // (4,3,1,2)  $\Leftrightarrow$  (X4,X3,X1,X2)
- // (4,3,2,1)  $\Leftrightarrow$  (X4,X3,X2,X1)

1) // (1, 3, 5, 7, 6, 4, 2) – поиск элементов перестановки

2) // (1, 3, 6, 7, 5, 4, 2) – перестановка элементов

3) // (1, 3, 6, 2, 4, 5, 7) – транспонирование



# Сортировка перебором перестановок

```
Const n=10;
Var a, p:array[1..n] of integer;
    i, j, k, r:integer;
Function sort:boolean; // проверка упорядоченности  
перестановки
Var ch:boolean;
Begin
    ch:=true;
    for i:=1 to n do
        if a[p[i]]>a[p[i+1]] then ch:=false ;
    sort:=ch
End;
Procedure print; // печать упорядоченной  
последовательности
Begin
    for i:=1 to n do
        write(a[pi], ' ')
End;
```

Begin

**// ВВОД ДАННЫХ И ИНИЦИАЛИЗАЦИЯ  
перестановки**

for i:=1 to n do begin

    a[i]:=random(100);

    p[i]:=i

end;

repeat // **проверка упорядоченности и вывод результата**

if sort then begin print; halt end;

j:=n; // 1) **(1,3,5,7,6,4,2)** – поиск элементов **перестановки**

while (j>0) and (p[j]<p[j-1]) do j-=1;

if j>0 then begin

k:=n;

while a[p[k]]<a[p[j]] do k-=1;

// 2) **(1,3,6,7,5,4,2)** – перестановка элементов

r:=a[p[k]]; a[p[k]]:=a[p[j]]; a[p[j]]:=r;

k:=(n-j) div 2; // 3) **(1,3,6,2,4,5,7)** – транспонирование

while k>0 do begin

r:=a[p[k+j]]; a[p[k+j]]:=a[p[n-k+1]]; a[p[n-k+1]]:=r

end

end

until j=0 // **условие завершения**

End.

# $N!$ $\Leftrightarrow$ время работы в сутках

$$5! = 120 \Leftrightarrow 3e-10$$

$$6! = 720 \Leftrightarrow 2e-9$$

$$7! = 5\,040 \Leftrightarrow 2e-8$$

$$8! = 40\,320 \Leftrightarrow 1.6e-7$$

$$9! = 362\,880 \Leftrightarrow 1.6e-6$$

$$10! = 3\,628\,800 \Leftrightarrow 1.8e-5 \text{ – секунда!}$$

$$11! = 39\,916\,800 \Leftrightarrow 2e-4$$

$$12! = 479\,001\,600 \Leftrightarrow 0.002$$

$$13! = 6\,227\,020\,800 \Leftrightarrow 0.04$$

$$14! = 87\,178\,291\,200 \Leftrightarrow 0.59 \text{ – пол дня!}$$

$$15! = 1307\,674\,468\,000 \Leftrightarrow 9.4$$

$$16! = 2e12 \Leftrightarrow 160.6$$

$$17! = 36e14 \Leftrightarrow 2895$$