

# КОМБИНАТОРИКА И ПРОГРАММИРОВАНИЕ

Учебное пособие и приложения в виде программ

Курилов И.А.

# СОДЕРЖАНИЕ:

- Что такое комбинаторика
- Факториал. Перестановки. Программы
- Размещение. Программы
- Сочетание. Программы
- Виды задач из ЕГЭ по информатике
- Вывод

# ЧТО ТАКОЕ КОМБИНАТОРИКА

- **Комбинаторика** (Комбинаторный анализ) — раздел математики, изучающий дискретные объекты, множества (**сочетания, перестановки, размещения и перечисления элементов**) и отношения на них (например, частичного порядка).
- **Комбинаторика связана с математикой** — алгеброй, геометрией, теорией вероятностей и имеет широкий спектр применения в различных областях знаний (например, в **генетике, информатике, статистической физике**).
- Термин «**комбинаторика**» был введён в математический обиход **Лейбницем**, который в 1666 году опубликовал свой труд «Рассуждения о комбинаторном искусстве».
- Иногда **под комбинаторикой понимают** более обширный раздел дискретной математики, включающий, в частности, **теорию графов (изучаем в информатике)**.

# ПРИМЕР ЗАДАЧ ПО КОМБИНАТОРИКИ

- Записать всевозможные двузначные числа, используя цифры 3, 5, 7. Подсчитать их количество.



*I-й метод (перебора).*

Решение:

35; 37; 53; 57; 73; 75; 33; 55; 77.

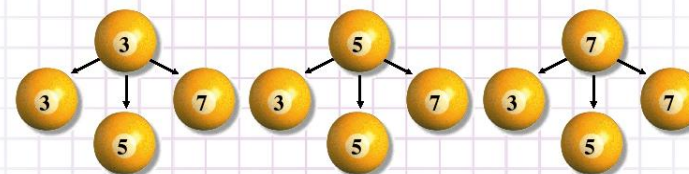
*II-й метод (таблица вариантов).*

Решение:

1-я цифра	2-я цифра		
	3	5	7
3	33	35	37
5	53	55	57
7	73	75	77

*III-й метод (дерево вариантов).*

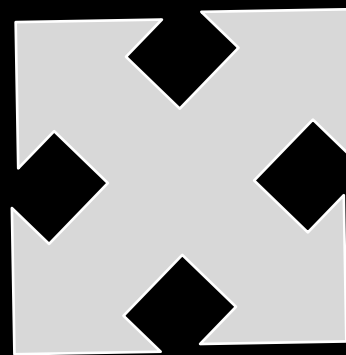
Решение:



33; 35; 37; 53; 55; 57; 73; 75; 77.

# РАЗНОВИДНОСТИ КОМБИНАЦИЙ

Перестановка



Размещение

Сочетание





# ФАКТОРИАЛ. ПЕРЕСТАНОВКИ

**Факториал числа  $n$**  (лат. *factorialis* — действующий, производящий, умножающий; обозначается  $n!$ , произносится эн факториал) — произведение всех натуральных чисел от 1 до  $n$  включительно

$$N! = 1 * 2 * 3 * 4 * \dots * N$$

$$5! = 1 * 2 * 3 * 4 * 5 = 120$$

В комбинаторике факториал натурального числа  $n$  интерпретируется как количество **перестановок -  $P$**  (упорядочиваний) множества из  $n$  элементов.

Например, для множества  $\{A, B, C, D\}$  из 4-х элементов существует  $4! = 24$  перестановки

ABCD	BACD	CABD	DABC
ABDC	BADC	CADB	DACB
ACBD	BCAD	CBAD	DBAC
ACDB	BCDA	CBDA	DBCA
ADBC	BDAC	CDAB	DCAB
ADCB	BDCA	CDBA	DCBA

Пример 1: В семье – шесть человек, а за столом в кухне – шесть стульев. В семье решили каждый вечер, ужиная, рассаживаться на эти шесть стульев по-новому. Сколько дней члены семьи смогут делать это без повторений?

Для удобства будем считать, что семья (бабушка, дедушка, мама, папа, дочь, сын) будет рассаживаться поочередно.

У бабушки – 6 вариантов выбора стульев.

У дедушки – 5 вариантов выбора стульев.

У мамы – 4 варианта выбора стульев.

У папы – 3 варианта выбора стульев.

У дочери – 2 варианта выбора стульев.

У сына – 1 вариант выбора стульев.

По правилу умножения:  $6 \times 5 \times 4 \times 3 \times 2 \times 1 = 720$  (дней).

Пример 2: В 10 классе в среду семь уроков: алгебра, геометрия, литература, русский язык, английский язык, биология и физкультура. Сколько можно составить вариантов расписания на среду?

Для алгебры – 7 вариантов расположения в расписании

Для геометрии – 6 вариантов

Для литературы – 5 вариантов и т.д.

По правилу умножения получаем  $= 7! = 5040$



Пример 3: Сколько различных четырёхзначных чисел, в которых цифры не повторяются, можно составить из цифр 0, 2, 4, 6?

Решение.

Из цифр 0, 2, 4, 6 можно получить  $P_4$  перестановок. Из этого числа надо исключить те перестановки, которые начинаются с 0, так как натуральное число не может начинаться с цифры 0. Число таких перестановок равно  $P_3$ . Значит, искомое число четырёхзначных чисел (без повторения цифр), которые можно составить из цифр 0, 2, 4, 6, равно  $P_4 - P_3 = 4! - 3! = 24 - 6 = 18$ .

# ФАКТОРИАЛ В ПРОГРАММИРОВАНИИ

## Нахождение факториала на языке Pascal

```
Var factorial: longint;  
    n, i: byte;  
begin  
    write('n = '); readln(n);  
    factorial := 1;  
    for i:=2 to n do  
        factorial := factorial * i;  
    writeln('n! = ', factorial);  
end.
```

## Нахождение факториала с помощью рекурсивной функции на языке Pascal.

```
var n: integer;  
function fact(n:integer):integer;  
  Begin  
    if n=1 then fact:=1 else fact:=fact(n-1)*n;  
  end;  
begin  
  write('vvedi chislo: ');  
  readln(n);  
  o:=fact(n);  
  writeln('otvet:', o);  
end.
```

## Программа вывода перестановок (уже для 4 элементов выглядит неэффективно)

```
const n=4; a:array[1..n] of integer =(1,2,3,4);
var i,j,k,l:integer;
begin
  writeln('Перестановки:');
  for i:=1 to n do
    for j:=1 to n do
      for k:=1 to n do
        for l:=1 to n do
          if (i<>j) and (i<>k)and (i<>l)and (j<>k)and (j<>l)and (k<>l) then write(a[i],a[j],a[k],a[l],'
');
        end.
      end.
    end.
  end.
```



# РАЗМЕЩЕНИЕ

Комбинаторике **размещением** (из  $n$  по  $k$ ) называется упорядоченный набор из  $k$  различных элементов из некоторого множества различных  $n$  элементов.

Пример: 1,3,2,5 — это 4-элементное размещение из 6-элементного множества 1,2,3,4,5,6

Пример: некоторые размещения элементов множества {1,2,3,4,5,6} по 2:

1,2 ; 1,3 ; 1,4 ; 1,5 ... 2,1 ... 2,3 ... 2,4 ... 2,6...

В отличие от сочетаний (смотрите далее), размещения учитывают порядок следования предметов. Так, например, наборы 2,1,3 и 3,2,1 являются различными, хотя состоят из одних и тех же элементов 1,2,3 (то есть совпадают как сочетания).

# ЗАДАЧИ

**Пример 1: Боря, Дима и Володя сели играть в «очко». Сколькими способами им можно сдать по одной карте? (колода содержит 36 карт)**

I способ -  $P_{36} = 34 \cdot 35 \cdot 36 = 42840$  способами можно раздать 3 карты игрокам.

II способ – по формуле размещений  $A_{36} = m! / (m-n)! = 36! / 33! = 42840$

III способ\* -  $C_{36} = m! / (m-n)! \cdot n! = 36! / 33! \cdot 3! = 7140$  способами можно извлечь 3 карты из колоды

Теперь давайте рассмотрим, какую-нибудь одну из семи тысяч ста сорока комбинаций, например: король пик, 9 червей, 7 червей. Выражаясь комбинаторной терминологией, эти 3 карты можно «переставить» между Борей, Димой и Володиёй  $P_3 = 3! = 6$

способами:

КП, 9Ч, 7Ч;  
КП, 7Ч, 9Ч;  
9Ч, КП, 7Ч;  
9Ч, 7Ч, КП;  
7Ч, КП, 9Ч;  
7Ч, 9Ч, КП.

И аналогичный факт справедлив для любого уникального набора из трёх карт. А таких наборов, не забываяем, мы насчитали 7140. Не нужно быть профессором, чтобы понять, что найденное количество комбинаций (\*сочетаний) следует умножить на шесть:

$$7140 \cdot 6 = 42840$$

Ответ: 42840 способов **ЭТОТ СПОСОБ** через сочетания **ДОСТАТОЧНО ЕМКИЙ!** Про сочетания смотрите дальше.



# ЗАДАЧИ

**Пример 2: Сколько существует вариантов распределения трех призовых мест, если в розыгрыше участвуют 7 команд?**

*I способ* -  $P_{36} = 7 * 6 * 5 = 210$  вариантов тройки лучших команд.

*II способ* – по формуле размещений  $A_{36} = m! / (m-n)! = 7! / 4! = 210$

*2 способ сводится к первому!*

$$7! / 3! = 7 * 6 * 5 * 4 * 3 * 2 * 1 / 4 * 3 * 2 * 1 = 7 * 6 * 5$$

# ПАСКАЛЬ ПРИМЕРЫ

## **Число размещений из $n$ элементов по $k$ элементов**

```
var i,r,k,n:integer;
begin
  writeln('k of n');
  readln(k,n);
  r:=1;
  for i:=1 to k do
    r:=r*(n-i+1);
  writeln(r);
end.
```

Можно также с использованием специальной формулы размещений.

- **Фрагмент программы вывода размещений (1: С повторениями\*, 2: Без повторений)**

```
1:for i:=1 to n do
```

```
  for j:=1 to n do begin
```

```
    write(a[i],a[j],' '); writeln(b[i],b[j]);
```

```
  end;
```

```
2:for i:=1 to n do
```

```
  for j:=1 to n do
```

```
    if i<>j then begin write(a[i],a[j],' '); writeln(b[i],b[j]); end;
```

Пояснения\*: Мы рассматривали задачи согласно обычной теории - размещения без повторений, так как имеются ввиду задачи с размещением групп людей ...  
**Но имеет место и задачи переборы комбинаций, например, цифр из 2, ... в этом случае назовем такие размещения с повторениями!**



# СОЧЕТАНИЕ

- В комбинаторике **сочетанием** из  $n$  по  $k$  называется набор  $\{k\}$  элементов, выбранных из данного множества, содержащего  $\{n\}$  различных элементов. **Наборы, отличающиеся только порядком следования элементов (но не составом), считаются одинаковыми, этим сочетания отличаются от размещений.**
- Так, например, наборы (3-элементные сочетания, подмножества,  $\{k=3\}$ )  $\{2, 1, 3\}$  и  $\{3, 2, 1\}$  6-элементного множества  $\{1, 2, 3, 4, 5, 6\}$  ( $\{n=6\}$ ) являются одинаковыми (в то время как размещения были бы разными) и состоят из одних и тех же элементов  $\{1, 2, 3\}$ .
- В общем случае число, показывающее, сколькими способами можно выбрать  $\{k\}$  элементов из множества, содержащего  $\{n\}$  различных элементов, стоит на пересечении  $\{k\}$ -й диагонали и  $\{n\}$ -й строки треугольника Паскаля.

# ЗАДАЧИ

**Пример 1: В ящике находится 15 деталей. Сколькими способами можно взять 4 детали?**

Здесь, конечно же, не нужно ворочать огромные числа.  $11! = 39916800$   $15! = 1307674368000$   
В похожей ситуации я советую использовать следующий приём: в знаменателе выбираем наибольший факториал (в данном случае  $11!$ ) и сокращаем на него дробь. Для этого числитель следует представить в виде  $15! = 11! * 12 * 13 * 14 * 15$

$$C_{36} = \frac{m!}{(m-n)! * n!} = \frac{11! * 12 * 13 * 14 * 15}{11! * 4!} = \frac{12 * 13 * 14 * 15}{24} = 1365$$

Ответ: 1365 способов

**Пример 2: Чемпионат России по шахматам проводится в один круг. Сколько играется партий, если участвуют 18 шахматистов?**

**Первый способ.** Каждый участник должен сыграть 17 партий, в каждой партии играют двое. Поэтому всего партий  $18 \cdot 17 : 2 = 153$ .

**Второй способ\*.** В каждой партии разыгрывается одно очко. Предположим, что все партии закончатся вничью. Тогда каждый участник наберет  $17 : 2 = 8,5$  очков. А всего очков, а значит, и партий –  $18 \cdot 8,5 = 153$ .

# ПОДСЧЕТ СОЧЕТАНИЙ

**Число размещений из  $n$  элементов по  $k$  элементов**

```
var i,s:longint; n,k:integer;
begin
  writeln('k of n');
  readln(k,n);
  s:=1;
  if k=0 then s:=1
    else for i:=1 to n-k do s:=s*(k+i) div i;
  writeln(s);
end.
```

Можно также с использованием специальной формулы сочетаний.



• **Фрагмент программы вывода сочетаний (1: С повторениями\*, 2: Без повторений)**

```
1:for i:=1 to n do
  for j:=i to n do begin
    write(a[i],a[j],' '); writeln(b[i],b[j]);
  end;
```

```
2:for i:=1 to n-1 do
  for j:=i+1 to n do begin
    write(a[i],a[j],' '); writeln(b[i],b[j]);
  end;
```

Пояснения\*: (Аналогично размещениям) Мы рассматривали задачи согласно обычной теории - сочетания без повторений, так как имеются ввиду задачи с размещением групп людей ... **Но имеет место и задачи переборы комбинаций, например, цифр из 2, ... в этом случае назовем такие сочетания с повторениями!**



## Полный листинг программы «Размещения и сочетания»

```
program kombin;  
  const n=4;  
  var a:array[1..n] of integer; b:array[1..n] of string;  
      i,j:integer; q,w:byte;  
begin  
  for i:=1 to n do a[i]:=i-1;  w:=64; for i:=1 to n do b[i]:=chr(w+i);  
  writeln('Выберите, что хотите получить:');  
  writeln('1:размещения с повторениями');  
  writeln('2:размещения без повторений');  
  writeln('3:сочетания с повторениями');  
  writeln('4:сочетания без повторений');  
  readln(q);  
  case q of  
    1:for i:=1 to n do  
      for j:=1 to n do begin write(a[i],a[j],' '); writeln(b[i],b[j]); end;  
    2:for i:=1 to n do  
      for j:=1 to n do if i<>j then begin write(a[i],a[j],' '); writeln(b[i],b[j]); end;  
    3:for i:=1 to n do  
      for j:=i to n do begin write(a[i],a[j],' '); writeln(b[i],b[j]); end;  
    4:for i:=1 to n-1 do  
      for j:=i+1 to n do begin write(a[i],a[j],' '); writeln(b[i],b[j]); end;  
  end;  
end.
```



# ЗАДАЧИ ИЗ ЕГЭ ПО ИНФОРМАТИКИ (№10)

**1. Все 5-буквенные слова, составленные из букв В, Е, К, Н, О, записаны в алфавитном порядке и пронумерованы. Вот начало списка:**

1. ВВВВВ
2. ВВВВЕ
3. ВВВВК
4. ВВВВН...

Заменим буквы В, Е, К, Н, О на 0, 1, 2, 3, 4 соответственно (для них порядок очевиден – по возрастанию).

Выпишем начало списка, заменив буквы на цифры: 1. 00000

2. 00001

3. 00002

4. 00003

Полученная запись есть числа, записанные в пятеричной системе счисления в порядке возрастания. Первое слово, начинающееся с "О" – 40000 переведем его в десятичную:

$$4 \cdot 5^4 + 0 \cdot 5^3 + 0 \cdot 5^2 + 0 \cdot 5^1 = 2500.$$

Не забудем о том, что есть слово номер 1, записывающееся как 0, а значит, 2500 – число, соответствующее номеру 2501.

Ответ: 2501.

# ЗАДАЧИ ИЗ ЕГЭ ПО ИНФОРМАТИКИ (№10)

2. Некоторый алфавит содержит 4 различных символа. Сколько трехбуквенных слов можно составить из символов этого алфавита, если символы в слове могут повторяться?

Пояснение:

Если в алфавите  $M$  символов, то количество всех возможных «слов» (сообщений) длиной  $N$  равно:

$$Q = M^N$$

$N=3$ ,  $M=4$ . Следовательно,  $Q = 4^3 = 64$ .

Ответ: 64

# ЗАДАЧИ ИЗ ЕГЭ ПО ИНФОРМАТИКИ (№10)

**3. Сколько существует различных символьных последовательностей длины 5 в четырёхбуквенном алфавите {А, С, G, T}, которые содержат ровно две буквы А?**

1. Рассмотрим различные варианты слов из 5 букв, которые содержат две буквы А и начинаются с А:

АА\*\*\*      А\*А\*\*      А\*\*А\*      А\*\*\*А

Здесь звёздочка обозначает любой символ из набора {С, G, T}, то есть один из трёх символов.

Итак, равно  $3^3 = 27$

Всего 4 шаблона, они дают  $4 \cdot 27 = 108$  комбинаций

2. Теперь рассматриваем шаблоны, где первая по счёту буква А стоит на второй позиции, их всего три:

\*АА\*\*      \*А\*А\*      \*А\*\*А

они дают  $3 \cdot 27 = 81$  комбинацию

3. Два шаблона, где первая по счёту буква А стоит на третьей позиции:

\*\*АА\*      \*\*А\*А

они дают  $2 \cdot 27 = 54$  комбинации

4. Один шаблон, где сочетание АА стоит в конце

\*\*\*АА

они дают 27 комбинаций

5. Всего получаем  $(4 + 3 + 2 + 1) \cdot 27 = 270$  комбинаций!



# КОНЕЦ

## Вывод:

- Комбинаторика и программирование неравно связаны в предмете информатика.
- Это задачи из материалов экзаменов, её знание имеет важное в олимпиадном программировании.
- Самое важным является практическое применение комбинаторики в программировании **для решения технических задач при автоматизации расчетов количества возможных ситуаций.**

