**Ларина Элла Семёновна**
*Кандидат физико-математических наук, доцент, лаборатория по работе с одарёнными детьми МФТИ.*

# Незаслуженно забытый двоичный перебор

Комбинаторные задачи – неизменные спутники всех олимпиад по программированию. Но вряд ли в школьном курсе информатики ребята проходят комбинаторику – уж очень ограничено время, отведённое на изучение программирования! В статье рассматривается один из методов получения комбинаторных групп – при помощи двоичного перебора. Данный метод предполагает работу с одномерными массивами, поэтому, если вы изучали на уроках информатики типовые алгоритмы обработки одномерных массивов, то, вооружившись полученными знаниями, у вас появится шанс успешного выступления на олимпиаде!

Для тех, кому не знакомо само понятие «Комбинаторика», советую положиться на интуицию и вдуматься в смысл этого слова: комбинаторика – комбинирование – комбинации. Женской половине читательской аудитории не нужно сильно напрягаться; понятно, что из трёх шарфиков синего, красного и зелёного цветов можно получить ровно семь новых аксессуаров:

- повязать на шею синий шарфик;
- накинуть шарфик красного цвета;
- зелёный шарф непринуждённо свесить через плечо;
- следующий цветовой вариант можно получить переплетением красного и зелёного шарфов;
- а теперь синий с зелёным;
- и синий с красным – неплохо!
- а что если все три вместе? Стильно!

Дай девушкам волю, так и ножницы пойдут в ход! Но, предположим, что шарфики резать нельзя. В таком случае мы перебрали *всевозможные* варианты комбинаций разных шарфиков и по отдельности, и друг с другом.

Для юношей, занимающихся спортом, пожалуй, будет интересен следующий пример: на штангу нужно повесить диски (гири) массами n1, n2, n3. Какие разнообразные варианты получения нагрузки имеются? Правильно: {n1}, {n2}, {n3}, {n1 + n2}, {n1 + n3}, {n2 + n3}, {n1 + n2 + n3}!

Ассоциативный «мостик» у нас найдётся для всех:

а) «Жадины», для вас задачка: есть монеты, достоинством 1 руб., 2 руб., 5 руб., 10 руб., представьте нам всевозможные группировки монет друг с другом.

б) Для увлеченных наукой, эдаких математиков-«ботанов», пример совершенно сухой и абстрактный: дано множество предметов в количестве $N$. Допустим, $N = 3$, множество {A, B, C}. Необходимо сформировать различные группы элементов, выбираемых из исходного множества. Количество элементов в выборке от 1 до $N$.

На этом примере и остановимся – ведь уже и так понятно, что, независимо от видов предметов (разноцветные шарфики, монеты, гири), нас интересует только их количество ($N$). Внимательный читатель уже знает, что при $N = 3$ всевозможных комбинаций предметов (элементов исходного множества) будет ровно 7. А если $N$ более трёх – чему равно количество комбинаций? И каковы будут эти группы (из каких элементов состоять)?

При $N = 2$, 3 перечислить всевозможные вариации нетрудно, а вот если количество элементов в исходном множестве будет больше, шанс запутаться велик! Что ж, настало время познакомиться с одним из методов формирования комбинаторных объектов, с помощью которого вероятность ошибки сведётся к нулю.

Составим таблицу, в шапке которой располагаются элементы исходного множества, а содержимое состоит из ноликов и единиц. Единица будет указывать на «выбранность» соответствующего элемента (стоящего в шапке этого столбца), ноль – на «невыбранность».

| A | B | C |
|---|---|---|
| 0 | 0 | 1 |
| 0 | 1 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 0 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |
| 1 | 1 | 1 |

Посмотрите внимательно на табличные данные и ответьте:

- Во-первых, всевозможные ли комбинации ноликов и единичек друг с другом представлены в таблице?
- И во-вторых, случайно ли расположение ноликов и единиц или наблюдается некая система получения очередной строки таблицы?

**Ответ:**

- Да, комбинации в таблице приведены всевозможные;
- Нет, строки таблицы сформированы не случайным образом, а соответствуют числам натурального ряда в двоичной системе счисления (отстутствует лишь комбинация «000»).

Очевидно, что *подряд* идущие числа от 001 до 111 дадут нам картину всевозможных разнообразных комбинаций ноликов и единичек друг с другом. В том, что они идут подряд, и заключается система получения комбинаторных групп, названная «Двоичным перебором».

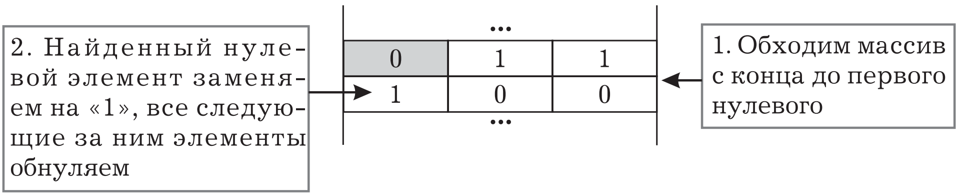
Итого, в нашем примере сформированы группы: {C}, {B}, {B, C}, {A}, {A, C}, {A, B}, {A, B, C}, количество групп равно $2^n - 1$.

Приступим к программной реализации задачи. Элементы исходного множества {A, B, C} будем хранить в массиве X:

| 1 | 2 | 3 |
|---|---|---|
| A | B | C |

Для выбора элементов из исходного множества необходимо на каждом шаге получать новый двоичный код (на единицу больше предыдущего).

- Первый вариант получения нового двоичного кода основывается на переводе счётчика цикла $i$ (работающего от 1 до $2^n - 1$) из десятичной в двоичную систему счисления.
- Второй вариант получения очередного двоичного кода заключается в следующем: ищем в массиве двоичных кодов **d** последний нулевой элемент, заменяем его на единицу и обнуляем все следующие за ним элементы (этот метод называется лексикографическим порядком):




Программная реализация алгоритма (на языке программирования Паскаль), базирующегося на получении нового двоичного кода методом лексикографического порядка, приведена ниже:

```
const nn=10;
var x,d: array [1..nn] of integer;
    i,n,k,j,z,st: integer;
begin
    writeln ('количество элементов'); readln (n);
    for i:= 1 to n do
       begin
       writeln ('введите элемент'); readln (x[i]);
       end;
    {=формирование двоичного кода===}
    st:=1;
    for i:=1 to n do
       st:=st*2;
    for i:= 1 to (st-1) do
       begin
       for j:= 1 to n do
            if d[j]= 0 then k:= j;
       for z:= k to n do
            d[z]:=0;
       d[k]:=1;
       {=печать элементов========}
       for j:= 1 to n do
            if d[j]<>0 then write (x[j]);
       writeln;
       end;
end.
```

А если предположить, что каждый элемент из исходного набора может повторяться во вновь созданной комбинаторной группе не единожды (от 0 до $n$ раз)? Допустим, что синий шарфик можно:

- вообще не повязывать на шею;
- повязать один шарф;
- повязать, скрутив жгутиком, два шарфика;
- повязать три шарфа;
- и т.д.

В таком случае необходимо организовать $n$-ичный перебор (сформировать натуральный ряд чисел в $n$-ичной системе счисления).

А теперь, если всё вышеизложенное понятно, попробуем решить несколько усложнённых задач, в которых сформированные комбинаторные группы объектов необходимо подвергнуть некоторому анализу. Например, не просто представить все варианты выборки монет из набора (достоинством 1, 2, 5, 10 рублей), но и проанализировать – можно ли выбранными монетами расплатиться за товар стоимостью Y рублей?

Или ещё задачи:

- Даны гири массами M1, M2, M3, M4. Как можно взвесить предмет массой X?
- Даны $N$ чисел. Выделите из них группы, содержащие от 1 до $N$ элементов, каждая из которых имеет сумму X.

В приведённой ниже программе цветом выделен фрагмент суммирования стоимостей выбранных монет и сравнения этой итоговой суммы со стоимостью товара:

```
const nn=10;
var x,d: array [1..nn] of integer;
  ii,i,n,y,j,z,st,k,sum: integer;
begin
  writeln ('введите с чем сравнивать');
  readln (y);
  writeln ('введите количество элементов');
  readln (n);
  for i:= 1 to n do
       begin
       writeln ('введите элемент'); readln (x[i]);
       end;
  {=вычисление количества возможных комбинаций=}
  st:=1;
  for i:=1 to n do
       st:=st*2;
  {==================================}
  for i:= 1 to (st-1) do
       begin
       {=получение следующего двоичного кода===}
       for j:= 1 to n do
            if d[j]= 0 then k:= j;
       for z:= k to n do
            d[z]:=0;
       d[k]:=1;
       {================================}
       sum:= 0;
       for j:= 1 to n do
            if d[j] <> 0 then sum:= sum + x[j];
       if sum = y then
```

```
                {=====вывод результата=========}
                for ii:= 1 to n do
                    if d[ii] <> 0 then write (x[ii]);
        writeln;
        end;
end.
```

Вот и всё. Хотя как-то грустно ставить точку и расставаться со столь интересной темой. Может быть, продолжим? Попробуйте самостоятельно:

- запрограммировать получение нового двоичного кода, основываясь на переводе счетчика цикла $i$ из десятичной в двоичную систему счисления.
- организовать $n$-ичный перебор (сформировать натуральный ряд чисел в $n$-ичной системе счисления).

## Калейдоскоп Калейдоскоп Калейдоскоп

### Высказывания классиков науки

На моих глазах наука превратилась в дело государственной важности… Она стала неотъемлемой и наиболее важной частью нашей цивилизации, а научная деятельность непосредственно влияет на развитие цивилизации.

*М. Борн*

Общая культура – это то, что позволяет человеку чувствовать всей душой солидарность с другими во времени и пространстве – как с людьми своего поколения, так и с ушедшими поколениями и с поколениями грядущими.

*П. Ланжевен*

Наука представляет собой внутренне единое целое. Её разделение на отдельные области обусловлено не столько природой, сколько ограниченностью способности человеческого познания. В действительности существует непрерывная цепь от физики и химии через биологию и антропологию к социальным наукам, цепь, которая ни в одном месте не может быть разорвана лишь по произволу.

*М. Планк*

Умение из частностей доходить до вероятно справедливого, а тем паче до достоверного или несомненно истинного и представляет существо научной самостоятельности.

*Д.И. Менделеев*