



Модуль 1 Перебор

Лекция 1.2 Рекурсивный перебор на примере

На прошлой лекции приводилась функция `rec` для перебора всех последовательностей длины n чисел от 1 до m (см. рис. 1). На этой лекции мы разберём по шагам работу этой функции на примере.

```
void rec(int idx)
{
    if (idx == n)
    {
        out();
        return;
    }
    for (int i = 1; i <= m; i++)
    {
        a[idx] = i;
        rec(idx + 1);
    }
}
```

Рис. 1. Решение задачи о переборе последовательностей

Пусть $n = 2$, $m = 3$, то есть требуется перебрать последовательности длины 2 из чисел 1, 2 и 3. Сначала мы заходим в функцию `rec` с $\text{idx} = 0$. Очевидно, условие не выполнится:

if ($\text{idx} == n$)

Поэтому мы сразу переходим к циклу.

for (**int** $i = 1$; $i \leq m$; $i++$)

На первой итерации цикла $i = 1$. Значение $a[0]$ присваивается 1:

$a[\text{idx}] = i$;

Переходим на новый уровень рекурсии.

`rec(idx + 1);`

Теперь $\text{idx} = 1$. Снова условие не выполнится:

if ($\text{idx} == n$)

Мы снова в цикле, и $i = 1$.

for (**int** $i = 1$; $i \leq m$; $i++$)

Присваиваем $a[1] = 1$:

$a[\text{idx}] = i$;

Переходим на следующий уровень рекурсии.

`rec(idx + 1);`

Индекс idx равен 2, то есть n . Условие выполняется.

if ($\text{idx} == n$)

Выводится содержимое вектора: последовательность «1 1».

```
out();
```

Выполняется выход из рекурсии.

```
return;
```

Мы возвращаемся на предыдущий уровень, где $idx = 1$. Мы находимся в цикле, и значение i в данный момент равно 1. На следующей итерации оно увеличивается до 2.

```
for (int i = 1; i <= m; i++)
```

Выполняется присваивание $a[1] = 2$.

```
a[idx] = i;
```

Мы снова переходим на следующий уровень рекурсии.

```
rec(idx + 1);
```

Снова $idx = 2$.

```
if (idx == n)
```

Происходит вывод новой последовательности «1 2».

```
out();
```

И выход из рекурсии.

```
return;
```

Далее мы снова вернёмся на предыдущий шаг, заменим в ячейке $a[1]$ значение 2 на 3 и выведем последовательность «1 3». Выполнение цикла при $idx = 1$ завершится.

Затем мы вернёмся на самый первый уровень рекурсии, с $idx = 0$. В $a[0]$ заменим 1 на 2. Заметим, что в ячейке $a[1]$ значение 3, которое осталось с предыдущих шагов. На следующем уровне рекурсии оно заменится сначала на 1, потом на 2, потом на 3. Будут выведены следующие три последовательности: «2 1», «2 2», «2 3». Затем то же самое повторится с первым числом, равным 3. В результате мы получим все возможные последовательности длины 2 из чисел 1, 2 и 3.

Работу функции `rec` можно представить как обход дерева на рис. 2. В каждой вершине дерева записана последовательность. В вершине на нулевом уровне — пустая последовательность. При переходе от вершины к её потомкам к последовательности добавляется число 1, 2 или 3. Наша задача — вывести все последовательности на втором уровне, состоящие из двух чисел. Функция `rec` реализует обход вершин этого дерева. Индекс idx соответствует уровню вершины.

Таким образом, мы разобрали код программы, которая решает задачу о переборе последовательностей, и её работу на примере.

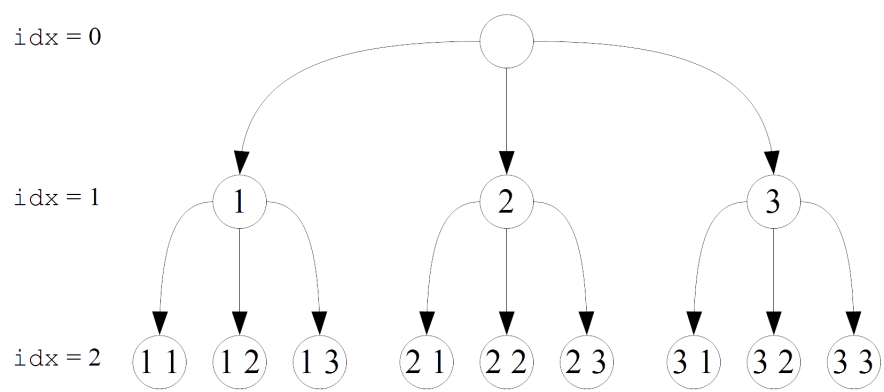


Рис. 2