Модуль 1 Перебор
Лекция 1.5 Разбиение числа на слагаемые

Эта лекции посвящена перебору разбиений заданного числа на слагаемые. Для определённости будем считать разбиения, которые отличаются только порядком слагаемых, одинаковыми. Например, $1 + 3 + 6$ и $6 + 3 + 1$ — одно и то же разбиение. Тогда можно перебирать только разбиения, в которых слагаемые идут в порядке неубывания. Например, для числа 5 получится 7 таких разбиений:

$$5 = 1 + 1 + 1 + 1 + 1,$$
$$5 = 1 + 1 + 1 + 2,$$
$$5 = 1 + 1 + 3,$$
$$5 = 1 + 2 + 2,$$
$$5 = 1 + 4,$$
$$5 = 2 + 3,$$
$$5 = 5.$$

Будем решать задачу при помощи рекурсивного перебора (см. рис. 1).

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

Рис. 1. Перебор разбиений числа на слагаемые

Так же, как в прошлых задачах, будем помещать элементы очередного разбиения в вектор a.

В рекурсивную функцию передаются индекс текущего элемента в векторе idx, накопленная сумма sum и последнее добавленное слагаемое last, числа меньше которого мы использовать уже не можем.

void rec(**int** idx, **int** sum, **int** last)

Перебираем в цикле новое слагаемое i.

for (**int** i = last; i <= n - sum; i++)

Цикл идёт от предыдущего слагаемого last до разности (n - sum). Здесь n — это данное нам число, для которого мы строим разбиения; sum — сумма, которую мы уже набрали; (n - sum) — сумма, которую нам осталось набрать. В цикле новое слагаемое помещается в ячейку a[idx].

a[idx] = i;

И происходит переход на следующий уровень рекурсии.

rec(idx + 1, sum + i, i);

При этом к сумме прибавляется новое слагаемое i, и в качестве last передаётся i.

Нам нужно выходить из рекурсии, когда набранная сумма стала равна n:

if (sum == n)

В этом случае мы выводим содержащееся в векторе a разбиение (первые idx его элементов) при помощи функции out, в которую в качестве параметра передается idx.

out(idx);

Затем выходим из рекурсии.

return;

В результате функция rec выведет все интересующие нас разбиения числа на слагаемые. Таким образом, мы убедились, что рекурсивный перебор можно использовать для достаточно разных комбинаторных объектов.