

P1157 组合的输出

题目描述

排列与组合是常用的数学方法，其中组合就是从 n 个元素中抽出 r 个元素(不分顺序且 $r \leq n$)，我们可以简单地将 n 个元素理解为自然数 $1, 2, \dots, n$ 从中任取 r 个数。

现要求你输出所有组合。

例如 $n=5, r=3$ 所有组合为：

123, 124, 125, 134, 135, 145, 234, 235, 245, 345

输入格式

一行两个自然数 n, r ($1 < n < 21, 0 \leq r \leq n$)

输出格式

所有的组合，每一个组合占一行且其中的元素按由小到大的顺序排列，每个元素占三个字符的位置，所有的组合也按字典顺序。

****注意哦！输出时，每个数字需要3个场宽。**

解析

如何筛选子集？

子集问题最常用的方法之一就是二进制法，我们先来回忆一下左移和右移。

给定数字 $N=3$ ，我们将数字1左移 N 位后，其二进制形式就变成了1000，再将其减1，就变成了111，我们可以把111看成 $\{1, 2, 3\}$ 这样的集合，同理101可以看成子集 $\{1, 3\}$ 。

根据这个性质，我们可以轻松化解大部分的子集类问题。

我们先将给定的 N 个数字的义为数字1左移 N 位的变量 U ，之后，再将 U 减1得到二进制数字 S ， S 就是包含给定 N 个数字全体的二进制形式，即 N 个1的形式。然后对 S 遍历减1，并通过内置函数__builtin_popcount统计当前的二进制数字包含1的个数是否与指定筛选的子集数目相等。

最后，我们通过对 U 进行遍历右移1位，并与 S 进行与运算，确定到底存在哪一位数字，即可得出最终结果。

编码

```
#include <bits/stdc++.h>

using namespace std;
int a[22] = {0};

int main(int argc, char **argv) {
    int n, k;
    cin >> n >> k;
    //将n个数字存入数组，便于后面进行输出
    for (int i = 1; i <= n; i++) {
```

```

        a[i] = i;
    }
    //左移n位，转化成二进制形式
    int u = 1 << n;
    for (int s = u - 1; s >= 0; s--) {
        //判断当前的1与目标需求相同
        if (builtin_popcount(s) == k) {
            //此时u的形式为1000，右移之后即为100
            for (int j = 1; j <= n; j++) {
                int temp = u >> j;
                //如果s中在j的位置为1，则证明当前位置有数字
                //这里采取的是个取巧的思路，s的真正数值与a并没有直接关系，我们只是
                if (s & temp) {
                    printf("%3d", a[j]);
                }
            }
            cout << endl;
        }
    }
    return 0;
}

```

逻辑航线培优教育，信息学奥赛培训专家。

扫码添加作者获取更多内容。



📍取这个位置