

以下内容参考自北大版《组合数学》。

k部分拆数

分拆：将自然数n写成递降正整数和的表示。

$$\$ \$ n = r_1 + r_2 + \dots + r_k \quad \text{quad } r_1 \geq r_2 \geq \dots \geq r_k \geq 1 \$ \$$$

和式中每个正整数称为一个部分。

分拆数 p_n 自然数n的分拆方法数。

自0开始的分拆数：

n	0	1	2	3	4	5	6	7
p_n	1	1	2	3	5	7	11	15

其中恰有k个部分的分拆，称为k部分拆数，记作 $p(n,k)$ 。

本题要求计算k部分拆数 $p(n,k)$ 。多组输入，其中n上界为10000，k上界为1000，对1000007取模。

显然 $p(n,k)$ 同时也是下面方程的解数：

$$\$ \$ n - k = y_1 + y_2 + \dots + y_k \quad \text{quad } y_1 \geq y_2 \geq \dots \geq y_k \geq 0 \$ \$$$

如果这个方程里面恰有j个部分非0，则恰有 $p(n-k,j)$ 个解。因此有和式：

$$\$ \$ p(n,k) = \sum_{j=1}^k p(n-k,j) \$ \$$$

相邻两个和式作差，得：

$$\$ \$ p(n,k) = p(n-1,k-1) + p(n-k,k) \$ \$$$

根据这个可以轻易地写出程序。

```
#include<stdio.h>
#include<string.h>

int p[10005][1005]; /* 将自然数n分拆为k个部分的方法数 */

int main()
{
    int n, k;
    while(~scanf("%d%d", &n, &k))
    {
        memset(p, 0, sizeof(p));
        p[0][0]=1;
        int i;
        for(i=1; i<=n; ++i)
        {
            int j;
            for(j=1; j<=k; ++j)
            {
                if(j < i)
                    p[i][j] = p[i-1][j];
                else
                    p[i][j] = (p[i-1][j] + p[i-j][j-1]) % 1000007;
            }
        }
    }
}
```

```
        if(i-j>=0)/*p[i-j][j]所有部分大于1*/
        {
            p[i][j]=(p[i-j][j]+p[i-1][j-1])%1000007; /*p[i-1][j-1]至少有一个部分为1。*/
        }
    }
    printf("%d\n",p[n][k]);
}
}
```

小结论一

生成函数：一种幂级数。各项的系数为数列中的对应项。

由等比数列求和公式，有：

$$\$ \$ 1/(1-x^k) = 1+x^k+x^{2k}+x^{3k}+\cdots \$ \$$$

$$\$ \$ 1+p_1 x+p_2 x^2+p_3 x^3+\cdots = \frac{1}{1-x} \frac{1}{1-x^2} \frac{1}{1-x^3} \cdots \$ \$$$

对于k部分拆数，生成函数稍微复杂。具体写出如下：

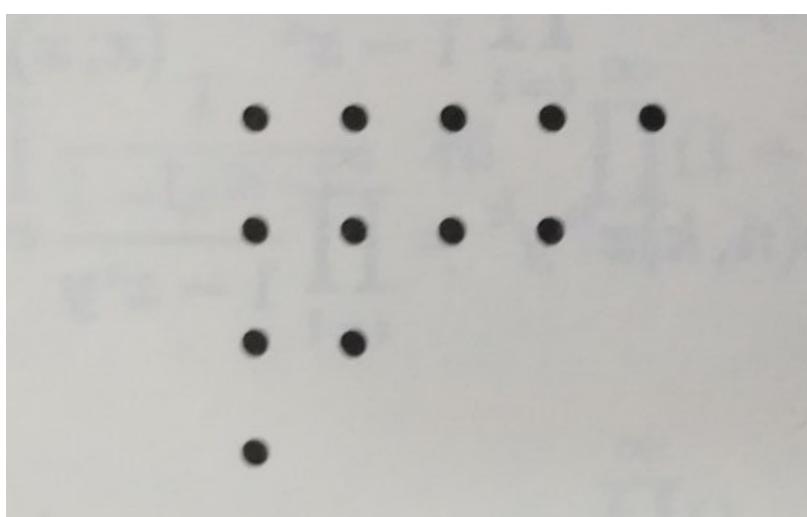
$$\$ \$ \sum_{n,k=0}^{\infty} \{p(n,k) x^n y^k\} = \frac{1}{1-xy} \frac{1}{1-x^2y} \frac{1}{1-x^3y} \cdots \$ \$$$

小结论二

Ferrers图：将分拆的每个部分用点组成的行表示。每行点的个数为这个部分的大小。

根据分拆的定义，Ferrers图中不同的行按照递减的次序排放。最长行在最上面。

例如：分拆 $12=5+4+2+1$ 的Ferrers图。



将一个Ferrers图沿着对角线翻转，得到的新Ferrers图称为原图的共轭，新分拆称为原分拆的共轭。显然，共轭是对称的关系。

例如上述分拆 $12=5+4+2+1$ 的共轭是分拆 $12=4+3+2+2+1$ 。

最大k分拆数：自然数n的最大部分为k的分拆个数。

根据共轭的定义，有显然结论：

最大k分拆数与k部分拆数相同，均为 $p(n,k)$

互异分拆数

互异分拆数 $\{pd\}_n$ 自然数n的各部分互不相同的分拆方法数 $\{Different\}$

互异偶部分拆数 $\{pe\}_n$ 自然数n的部分数为偶数的互异分拆方法数 $\{Even\}$

互异奇部分拆数 $\{po\}_n$ 自然数n的部分数为奇数的互异分拆方法数 $\{Odd\}$

因此有：

$$\{pd\}_n = \{pe\}_n + \{po\}_n$$

本题要求计算互异分拆数 $\{pd\}_n$ 多组输入，其中n上界为50000，对1000007取模。

同样地，定义互异k部分拆数 $p(n,k)$ 表示最大拆出k个部分的互异分拆，是这个方程的解数：

$$n = r_1 + r_2 + \dots + r_k \quad (r_1 > r_2 > \dots > r_k \geq 1)$$

完全同上，也是这个方程的解数：

$$n - k = y_1 + y_2 + \dots + y_k \quad (y_1 > y_2 > \dots > y_k \geq 0)$$

这里与上面不同的是，由于互异，新方程中至多只有一个部分非零。有不变的结论：恰有j个部分非0，则恰有 $p(n-k,j)$ 个解，这里j只取k或k-1因此直接得到递推：

$$p(n,k) = p(n-k,k-1) + p(n-k,k)$$

代码如下。代码中将后一位缩减了空间，仅保留相邻两项。

```
#include<stdio.h>
#include<string.h>

int pd[50005][2]; /* 将自然数n分拆为k个部分的互异方法数 */

int main()
{
    int n;
    while(~scanf("%d", &n))
    {
        memset(pd, 0, sizeof(pd));
        pd[0][0]=1;
        int ans=0;
        int j;
        for(j=1; j<350; ++j)
        {
            if(j>n)
                break;
            pd[j][0] = pd[j-1][0];
            pd[j][1] = pd[j-1][1];
            if(j>1)
                pd[j][1] += pd[j-2][0];
        }
        ans = pd[n][1];
        if(ans>350)
            ans = 350;
        printf("%d\n", ans);
    }
}
```

```
int i;
for(i=0;i<350;++i)
{
    pd[i][j&1]=0; /*pd[i][j]只与pd[][j]和pd[][j-1]有关*/
}
for(i=0;i<=n;++i)
{
    if(i-j>=0)/*pd[i-j][j]所有部分大于1*/
    {
        pd[i][j&1]=(pd[i-j][j&1]+pd[i-
j][(j-1)&1])%1000007; /*pd[i-j][j-1]至少有一个部分为1。 */
    }
}
ans=(ans+pd[n][j&1])%1000007;
printf("%d\n",ans);
}
```

小结论三

奇分拆数：自然数n的各部分都是奇数的分拆方法数。

有一个显然的等式：

$$\prod_{i=1}^{\infty} (1+x^i) = \frac{\prod_{i=1}^{\infty} (1-x^{2i})}{\prod_{i=1}^{\infty} (1-x^{2i-1})}$$

最左边是互异分拆数的生成函数，最右边是奇分拆数的生成函数。两者对应系数相同，因此，奇分拆数和互异分拆数相同，均为 p_n

分拆数

本题要求计算分拆数 p_n 多组输入，其中 n 上界为 50000，对 1000007 取模。

单独观察分拆数的生成函数的分母部分：

$$\prod_{i=1}^{\infty} (1-x^i)$$

将这部分展开，可以想到互异分拆，与互异分拆拆出的部分数奇偶性有关。

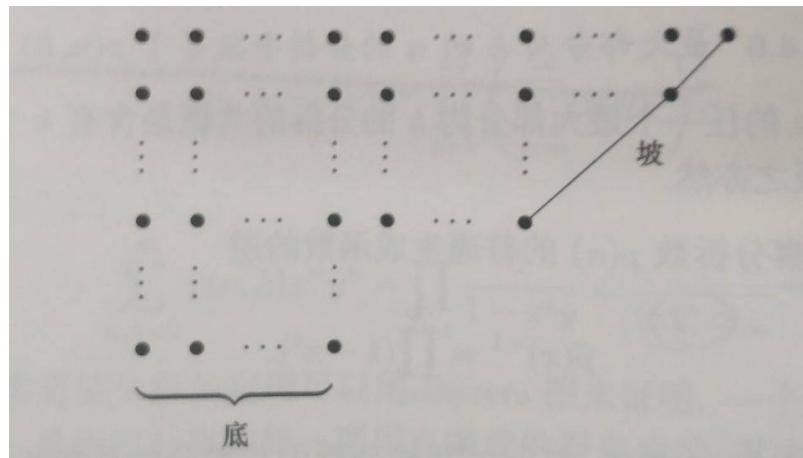
具体地，互异偶部分拆在展开式中被正向计数，互异奇部分拆在展开式中被负向计数。因此展开式中各项系数为两方法数之差。即：

$$\sum_{i=0}^{\infty} (\{pe\}_n - \{po\}_n) x^n = \prod_{i=1}^{\infty} (1-x^i)$$

接下来说明，多数情况下，上述两方法数相等，在展开式中系数为 0；仅在少数位置，两方法数相差 1 或 -1。

这里只能借助构造对应的办法。

画出每个互异分拆的Ferrers图。最后一行称为这个图的底，底上点的个数记为 b Bottom 连接最上面一行的最后一个点与图中某点的最长45度角线段，称为这个图的坡，坡上点的个数记为 s Slide



From:
<https://wiki.cvbbacm.com/> - CVBB ACM Team

Permanent link:
<https://wiki.cvbbacm.com/doku.php?id=2020-2021:teams:namespace:%E6%95%B4%E6%95%B0%E5%88%86%E6%8B%86%E9%97%AE%E9%A2%98&rev=1588750984>

Last update: 2020/05/06 15:43

