

P0J1306 【基础】

题目大意：

给出 N 件物品，要选 M 件，求有多少种选的方法。

输入：

有若干组测试数据，每一组有两个整数 N 、 M ($1 \leq N$ 、 $M \leq 100$)， $N=M=0$ 时测试数据结束。

输出：

每一组测试数据输出一行，格式为 “ $\%d$ things taken $\%d$ at a time is $\%d$ exactly.” 三个 $\%d$ 依次代表 N 、 M 和 $C(N, M)$ 。

题解：

原题明确给出了 $100!$ 是 159 位的，因此如果直接用组合数公式去算需要用高精度，很麻烦。但是题目明确告诉我们，结果可以用 `long` (C/C++) 或者 `longint` (Pascal) 类型存下来。由组合数学知识可知， $C(n, k) = C(n-1, k-1) + C(n-1, k)$ 。这很好理解：如果原来有 $i-1$ 个元素可以选，现在新增一个，那么选新增的那个元素就只能在原来的 $i-1$ 个元素里选 $j-1$ 个元素，或者完全在原来的 $i-1$ 个元素里选 j 个元素。同时应该注意，这个公式和杨辉三角的公式是一样的，以后我们将经常碰到。

P0J1942 【基础】

题目大意：

给出一个 $a*b$ 的矩阵 ($1 \leq a, b \leq 32$ 位的无符号整数能表达的最大范围)，求从左上角走到右下角有多少种走法，只能向右或者向下走。

输出：

有若干组测试数据。每一组测试数据有一行，有两个空格分隔的整数 a 和 b ，当 $a=b=0$ 时测试数据结束。

输出：

每一组测试数据输出一行，包含一个整数。

题解：

这题小学生都知道，走到某一个的方法总数等于走到其上一格和左一格的方法数之和。假设走到某个格子一共已经走了 n 格了，并且距离原来的起点已经向下或者向右的 k 格，那么一共的走法就是在 n 步里面选 k 步向下或者向右走，也就是 $c(n, k)$ 了，并且显然由上述得到 $C(n, k) = C(n-1, k-1) + C(n-1, k)$ ，和组合数公式是符合的。因此这里 $n=a+b$ ， $k=a$ 或者 b 。

POJ2084 【基础】

题目大意：

一个圆周上的 $2*n$ 个点两两连线，要求每个点必须与且仅与一个点相连，问有多少种方法。

输出：

有若干组测试数据，每一组测试数据有一行，包含一个整数，即 n 。 $n=-1$ 时测试数据结束。

输出：

每一组测试数据输出一行，包含一个整数，即连线的方法数。

题解：

对于 $2*n$ 个点，任意选择其中一个点，这个点能和与它相距偶数个点的点相连，而相连的这条线会把整个图形分成两个子问题，即得到递推公式：

$$f[n] = f[0]*f[n-1] + f[1]*f[n-2] + f[2]*f[n-3] + \dots + f[n-1]*f[0]。$$

这个公式被称为卡特兰数。有一个更简洁的递推公式： $f[n] = (4n-2)/(n+1)*f[n-1]$ ($n>1$)， $f[0]=1$ 。这个递推关系的解为递推关系的解为：

$$f[n] = c(2n, n)/(n+1)。$$

由于可以预见到数据范围将非常大，所以这题需要用高精度计算。并且为了不写高精度除法那么麻烦，我用了朴素的递归公式。

POJ3842 【基础】

题目大意：

给出 N 个个位数 ($1 \leq N \leq 10$)，这些数可能有相同的，问用这些数能组合出多少个素数 ($1 \leq \text{位数} \leq N$)。

输入：

第一行有一个整数 T，表示有 T 组测试数据。每一组测试数据有一行，包含若干个数字。

输出：

每一组测试数据输出一行，包含一个整数，即有多少个素数。

题解：

主要就是全排列和素数检验。素数检验可以先打一万以内的素数表，大于一万的再按照表里的进行分解检验。

POJ1289 【中等】

题目大意：

（原题意比较有意思，我直接把数学模型抽出来）已知一棵满 M 叉树，该树任意结点的值是其父结点值的 $1/(M+1)$ ，叶节点的值 1。已知根结点的值（记为 H）和叶结点的个数（记为 N）。求非叶结点的个数和所有结点值之和。

输入：

有若干组测试数据，每一组测试数据有一行，输出两个整数，即 H 和 N。N=H=0 时测试数据结束。

输出：

每一组测试数据输出一行，包含两个整数，即非叶结点的个数和所有结点值之和。

题解：

记给定树的深度为 D（根节点深度为 0），则由题意可以得出：1、 $M^D = N$ ；2、 $(M+1)^D = H$ 。枚举 D 进行判断，注意使用 pow 的时候要用 eps 进行精度控制而不能直接判断相等。然后由等比数列求和可以得到非叶节点个数 $= (1 - M^{(D-1)}) / (1 - M)$ 即 $(N - 1) / (M - 1)$ ，所有节点值之和我推了几次没写对公式以后就直接用了一个循环来算了。另外有一些特殊的取值情况需要考虑，比如 $n=1$ 、 $h=1$ 时直接输出 0 1， $n=1$ 、 $h \neq 1$ 的时候输出 1 0。

POJ1496 【中等】

题目大意：

给出一个字符串，求其在一个升序串的字典中的位置。这个升序的字典第 1 到 26 是 a 到 z，27~51 是 ab ac az，接下去是 bc bd bz xy xz yz abc abd

输入：

有若干组测试数据，每一组测试数据有一行，包含一个小写字母组成的字符串，长度不超过 5 个字符。

输出：

对于每一组测试数据，输出一行，一个整数，即其位置。如果给出的串不是升序的，直接输出 0。

题解：

给定一个长度，问有多少符合的字符串，相当于在 26 个字母里随意选长度个字符（不重复）然后按字典升序排列。因此有 $C(26, len)$ 种选法。由组合数公式可知和杨辉三角的递推是一样的。所以我们可以先把杨辉三角的表打出来。

对于一个长度为 len 的串，先检查是不是升序的，不是就直接输出。我们需要先求出长度比它短的串有多少个。这很好解决，因为直接查表求和 $c[26][i]$ 就可以了。然后枚举当前串的每一位：第一位从 a 开始，其余位从前一位的下一个字符开始，一直到当前的实际的字符为止，求后面的位数里有多少个符合要求的串。求和以后输出即可。

当然因为这一题数据范围实在太小，所以其实还可以直接先打表，然后查表输出。

POJ 1850 和这一题是完全一样的，不过是位数变大了因此不能打表了，而且只有一组测试数据。我给出一个两题通用的程序。

POJ1012【中等偏上】

题目大意：

有 k ($1 \leq k \leq 14$) 个好人和 k 个坏人，依好人-坏人的次序排成一个圈，第一个好人开始从 1 编号。现在给定一个整数 m ，先从 1 开始数，数到 m 的人自杀，他的编号的下一个好人开始从 1 进行数，数到 m 的人再自杀，然后重复这个流程。现在需要让所有的坏人都在好人死掉之前死掉，要求最小的 m 。

输入：

有若干组测试数据，每一组测试数据一行，有一个整数 k 。 $k=0$ 的时候测试数据结束。

输出：

每一组测试数据输出一行，包含最小的 m 。

题解：

这题是很经典的一题约瑟夫环问题。约瑟夫环是指 n 个人依次编号从 0 到 $n-1$ ，给出一个整数 m ，一开始从 0 开始报号，报到 $m-1$ 的人推出，下一个人从 0 开始继续报号，求最后留下来的是谁。如果用链表或者数组来模拟的话耗时会太大，因此有数学上的递归方法：

第一个人（编号为 $(m-1) \bmod n$ ）出列后，剩下的 $n-1$ 个人组成了一个新的约瑟夫环，原来的编号与后来的编号对应的是：（记 $k=m \bmod n$ ）

原编号： $k \ k+1 \ k+2 \ \dots \ n-2 \ n-1 \ 0 \ 1 \ 2 \ \dots \ k-2$

新编号： $0 \ 1 \ 2 \ \dots \ \dots \ \dots \ \dots \ n-2$

那么我们得到了一个新的约瑟夫环，如果我们知道在这个环中最后的赢家是谁，那么我们只需要将赢家的编号变换回去就可以了。设新环的赢家是 x ，那么易得原环中编号为 $x'=(x+k) \bmod n$ 。求新的约瑟夫环则可以通过求出一个出列的人再得到新的约瑟夫环，因此便可以递归得到答案。

记 $f[i]$ 为给定 m 且 $n=i$ 时约瑟夫环的胜利者编号，那么可以得到递推公式：

$f[1]=0$ ； $f[i]=(f[i-1]+m) \bmod i$ 。

在这题中我们做一个变形：

设 $f[i]$ 表示的当前第 i 轮序列中出局的人的编号（注意，这里的编号是当前子序列中的编号，当前的编号是 $1 \sim (n-i+1)$ ）， $f[i]=(f[i-1]+m-1) \bmod (n-i+1)$ 。（这里网上没有给出具体的解释，我也没有想得很清楚，先存疑。）

因此我们对 m 从 1 开始枚举，如果发现前 k 轮里有好人出局了就枚举下一个 m 。并且由于 k 的范围很小，所以可以把每一个 k 的答案保存下来，如果以后再遇到就直接查询。

POJ3252 【中等偏上】

题目大意：

我们规定一个数为 round number，如果这个数转化为二进制以后 0 的位数比 1 的位数多。现在给出 a 、 b ($1 \leq a < b \leq 2 \times 10^9$)，求 ab 之间有多少个 round number。

输入：

两个用空格分隔的整数 a 和 b 。

输出：

一行，一个整数，即有多少个 round number。

题解：

细节比较繁琐的一道题目，我参考了

<http://blog.csdn.net/zhengnanlee/article/details/9794625>这个题解才写出来的。