

百度松果菁英班刷题题解

2023Fall

- week1

1. 第一题——监狱

监狱有 n 个房间，每个房间关押一个犯人，有 m 种宗教，每个犯人会信仰其中一种。如果相邻房间的犯人的宗教相同，就可能发生越狱，求有多少种状态可能发生越狱。

答案对 1007 取模

- 格式:

- input: 输入只有一行两个整数，分别代表宗教数 m 和房间数 n
- output: 输出一行一个整数代表答案

- 解题思路:

显然对于 n 个房间来说，所有的可能情况是 m^n ，所以我们考虑所有房间的信仰都与隔壁不同的情况，发现其为 $m * (m - 1)^{n-1}$ ，所以答案就是 $m^n - m * (m - 1)^{n-1}$

- My Code:

```
mod = 1007
```

```
def quickPow(a, b):
    sum = 1
    while b:
        if b & 1:
            sum = sum * a % mod
        b >>= 1
        a = a * a % mod
    return sum

m, n = map(int, input().split())
ans1 = quickPow(m, n)
ans2 = (m % mod * quickPow(m - 1, n - 1)) % mod
ans = ans1 - ans2
if ans < 0:
    ans += mod
print(ans % mod)
```

2. 第二题——盒子与球

现有 r 个互不相同的盒子和 n 个互不相同的球，要将这 n 个球放入 r 个盒子中，且不允许有空盒子。请求出有多少种不同的放法。两种放法不同当且仅当存在一个球使得该球在两种放法中放入不同的盒子

- 格式:

- input: 输入只有一行两个整数，分别代表 n 和 r
- output: 输出一行一个整数代表答案

- 解题思路:

- 这是一个经典的dp问题，可以使用组合数学中的排列组合知识来解决。首先，我们需要确定每个盒子是否都有球。如果每个盒子都有球，那么问题可以转化为将 n 个球放入 r 个盒子中，允许有空盒子的情况。这个问题可以使用"stars and bars"方法来解决，结果为 $C(n-1, r-1)$ 。如果有一个或多个盒子是空的，我们可以从 r 个盒子中选择 k 个盒子作为空盒子，然后将剩下的球放入剩下的 $r-k$ 个盒子中。这个问题可以转化为将 n 个球放入 $r-k$ 个盒子中，允许有空盒子的情况。根据上面的讨论，这种情况的解法为 $C(n-1, r-k-1)$ 。最后，我们需要将所有可能的情况相加，即：总放法数 = $C(n-1, r-1) + C(n-1, r-2) + \dots + C(n-1, 0)$ 其中， $C(n, k)$ 表示从 n 个元素中选择 k 个元素的组合数，计算公式为 $C(n, k) = n! / (k! * (n-k)!)$

。

- My Code:

```

from math import factorial

def stirling_number(n, k):
    if k == 0 or n == 0:
        return 0
    elif k == n:
        return 1
    else:
        return stirling_number(n-1, k-1) + k * stirling_number(n-1, k)

def count_arrangements(n, r):
    if n < r:
        return 0
    else:
        return stirling_number(n, r) * factorial(r)

def main():
    #code here
    n, r = map(int, input().split())
    arrangements = count_arrangements(n, r)
    print(arrangements)

if __name__ == '__main__':
    main();

```

1. 第三题——square

在一个 $m \times n$ 的矩阵上，小码哥在左下角的顶点出现了，他只能沿着路径向上或者向右走，他的目标是“蠕动”到右上角的顶点，问他有多少路径可以选择。嗯，这个、这个、这个似乎地球人都知道怎么做，但是请注意，我有个条件没给呢！ m 和 n 现在的最大范围是50000，这可怎么办？仔细想想吧。

• 格式

- input: 只有一行，包含两个整数 m 和 n ，其均不小于4，上限均为50000.
- output: 由于最后的答案数目过大，所以只检查后100位，输出时每行十个数字，没空格间隔，共十行，如果答案位数没超过100位，则需要在空位上补0。

• 解题思路：

- 其实这个问题求解并不难，无非是 C_{m+n}^m ，问题在于这个数太大了， $m=n=500$ 的时候都会超过10的100次方，所以我们需要对结果进行取模，在debug的时候我发现python两个整型相除的结果是float，但是float没办法表示 10^{100} 这么大的数，所以我们要用//整除运算符去运算

- My Code:

```
m, n = map(int, input().split())

mn = m+n
fenzi = 1
fenmu = 1
for i in range(1,n+1):
    fenzi *= mn
    mn -= 1
    fenmu *= i
ans = (fenzi//fenmu)%10**100

# 输出结果
for i in range(10):
    for j in range(10):
        print(str(ans).zfill(100)[i*10+j], end='')
    print()
```

4. 第四题——找朋友

将n个人分成m组，每组至少一人，在比赛结束时，同一组的人两两之间都会成为朋友，不同分组的分组方案得到的朋友的对数不同。你的任务是求出最小和最大的朋友对数

- 格式

- input: 两个整数 n 和m
- output: 两个整数表示答案

- 解题思路:

- 我们简单举一个例子 (10, 4) 进行分析不难发现，最多的方案肯定是其他组全是1个人和一个组里人最多，最少的方案是尽量将m个组的人数平均,一个组里朋友的对数就是 C_x^2

- My Code:

```

def C_n2(n):
    ans = 1
    for i in range(1,3):
        ans *= n
        n -= 1
    ans //= 2
    return ans

def main():
    #code here
    n,m = map(int,input().split(' '))
    most = C_n2(n-m+1)
    if n%m == 0:
        least = C_n2(n//m)*m
    else:
        least = C_n2(n//m+1)*(n%m) + C_n2(n//m)*(m - n%m)
    print("%d %d"%(least,most))

if __name__ == '__main__':
    main();

```

1. 第五题——点餐

小码哥和他的两个朋友一起去吃饭，他们决定每个人先从菜单上选几道菜，然后点三个人都选中的菜。假设菜单中有 n 道菜，他们三人分别点了 a,b,c 道菜，小码哥想知道是否有可能不存在三个人都选中的菜。

- 格式
 - input: 一行，4 个以空格隔开的正整数 n,a,b,c 满足 $0 < a,b,c \leq n \leq 1000$
 - output: 一行，若可能不存在三个人都选中的菜输出 YES，否则输出NO
- 解题思路：
 - 其实，我们有时候只需要思考问题的反面， $n-a$ ， $n-b$ ， $n-c$ 道菜没有被点，只要这三部分加在一起大过餐馆菜数 n ，那就必然存在有道菜三人都没有点
- My Code：

```
def main():  
    #code here  
    n,a,b,c = map(int,input().split())  
    unselect_a = n-a  
    unselect_b = n-b  
    unselect_c = n-c  
    if unselect_a+unselect_b+unselect_c < n:  
        print("NO")  
    else:  
        print("YES")
```

```
if __name__ == '__main__':  
    main();
```