

组合数学

组合数学

有重复元素的全排列

找到选择字母数量和对应的个数

代码

有重复元素的全排列

首先考虑有重复元素的全排列,比如 3个红球, 2个黄球, 1个白球。他们的全排列结果是

$$x = \frac{A_6^6}{A_3^3 A_2^2 A_1^1}$$

分子是不重复元素的全排列,分母是每个相同元素的个数的全排列个数的乘积。

那么对于这道题,比如随机按键个数 $n = 6$, 每个字母最多按键 $k = 3$ 次。

有一种情况是选择3个字母, 分别按键3, 2, 1次, 对应的排列数计算顺序为

1. 从26个字母里面选择1个字母按3次, 从剩下的25个字母里面选择1个字母按2次, 从剩下的24个字母里面选择一个字母按1次
2. 选出字母来计算有重复元素全排列

对于步骤1, 计算公式为

$$y = C_{26}^1 C_{25}^1 C_{24}^1$$

对于步骤2, 计算公式为一开始介绍的公式, 将他们两个相乘就是这种情况下的结果。

找到选择字母数量和对应的个数

可以通过DFS搜索可能的选择字母的数量以及对敲了几次。通过列表保存, 列表长度为选择的字母的个数, 列表种每个元素为字母对应的使用次数

对于上面的例子, 搜索出来的数组为

```
1  [1, 1, 1, 1, 1, 1]
2  [1, 1, 1, 1, 2]
3  [1, 1, 1, 3]
4  [1, 1, 2, 2]
5  [1, 2, 3]
6  [2, 2, 2]
7  [3, 3]
```

对应的结果为(只写出了前5项)

$$\begin{aligned}
& C_{26}^6 \cdot \frac{A_6^6}{(A_1^1)^6} + \\
& C_{26}^4 \cdot C_{22}^1 \cdot \frac{A_6^6}{(A_1^1)^4 \cdot A_2^2} + \\
& C_{26}^3 \cdot C_{23}^1 \cdot \frac{A_6^6}{(A_1^1)^3 \cdot A_3^3} + \\
& C_{26}^2 \cdot C_{24}^2 \cdot \frac{A_6^6}{(A_1^1)^2 \cdot (A_2^2)^2} + \\
& C_{26}^1 \cdot C_{25}^1 \cdot C_{24}^1 \cdot \frac{A_6^6}{A_1^1 \cdot A_2^2 \cdot A_3^3} + \\
& \dots
\end{aligned}$$

代码

```

1  import math
2  from collections import Counter
3  from functools import lru_cache
4
5
6  @lru_cache(None)
7  def _comb(a, b):
8      return math.comb(a, b)
9
10
11 @lru_cache(None)
12 def _factorial(x):
13     return math.factorial(x)
14
15
16 class Solution:
17     def keyboard(self, k: int, n: int) -> int:
18         Ann = _factorial(n)
19         ret = 0
20         MOD = 10 ** 9 + 7
21         combined_ret = []
22
23         def helper(idx, cur_n, pre, pre_min):
24             if cur_n == 0:
25                 combined_ret.append(pre[:])
26                 return
27             if idx == 26:
28                 return

```

```

29         else:
30             for i in range(pre_min, min(k + 1, cur_n +
31 1)):
32                 helper(idx + 1, cur_n - i, pre + [i], i)
33
34     helper(0, n, [], 1)
35     for cmb in combined_ret:
36         cnt = Counter(cmb)
37         x = 1
38         y = 1
39         _len = 26
40         for ke, v in cnt.items():
41             x *= _comb(_len, v)
42             _len -= v
43             y *= _factorial(ke) ** v
44         ret += x * Ann // y
45
46     return ret % MOD

```