密码学大作业二：置乱的阶的分析

PB21071417 陈柯宇

实验原理

1. 对于N的最大阶，就是把N分解为一系列正整数相加，然后求这些数的最小公倍数，取其中的最大值便是N的最大阶
2. 求p(k)曲线，就是把N各种分解的阶求出来，统计各种阶对应分解的个数，然后再处理求概率就行。
3. 测评Logistic混沌映射，设置好mu和N，用随机数生成x_0，然后得到置乱求阶就行

实验过程

求N的最大阶

我们观察老师给出的例子，不难发现阶最大时，其分解中大多都是质数，所以要求最大阶，可以先分解为一系列质数相加，然后针对剩下的那个数，再进行讨论。

但是就在这时，我被同学告知有一个网站`oeis.org`，收藏了很多数列，其中就有最大阶的数列，于是直接拿来用了。

求p(k)曲线

这个问题，我思考了很久，并结合chatgpt的回答，得到了代码。

```
#  求给定N时，它的所有分解
def decompose(num):
    result = []
    decompose_helper(num, [], result)
    return result

def decompose_helper(num, current, result):
    if num == 0:
        result.append(current)
        return
    for i in range(1, num+1):
        if not current or i >= current[-1]:
            decompose_helper(num-i, current+[i], result)
```

然后使用Matplotlib库，画出曲线。

测评Logistic混沌映射

按照我的理解，就是设置好`mu`与`N`，然后随即改变`x_0`，得到不同的置乱，分析他们的阶

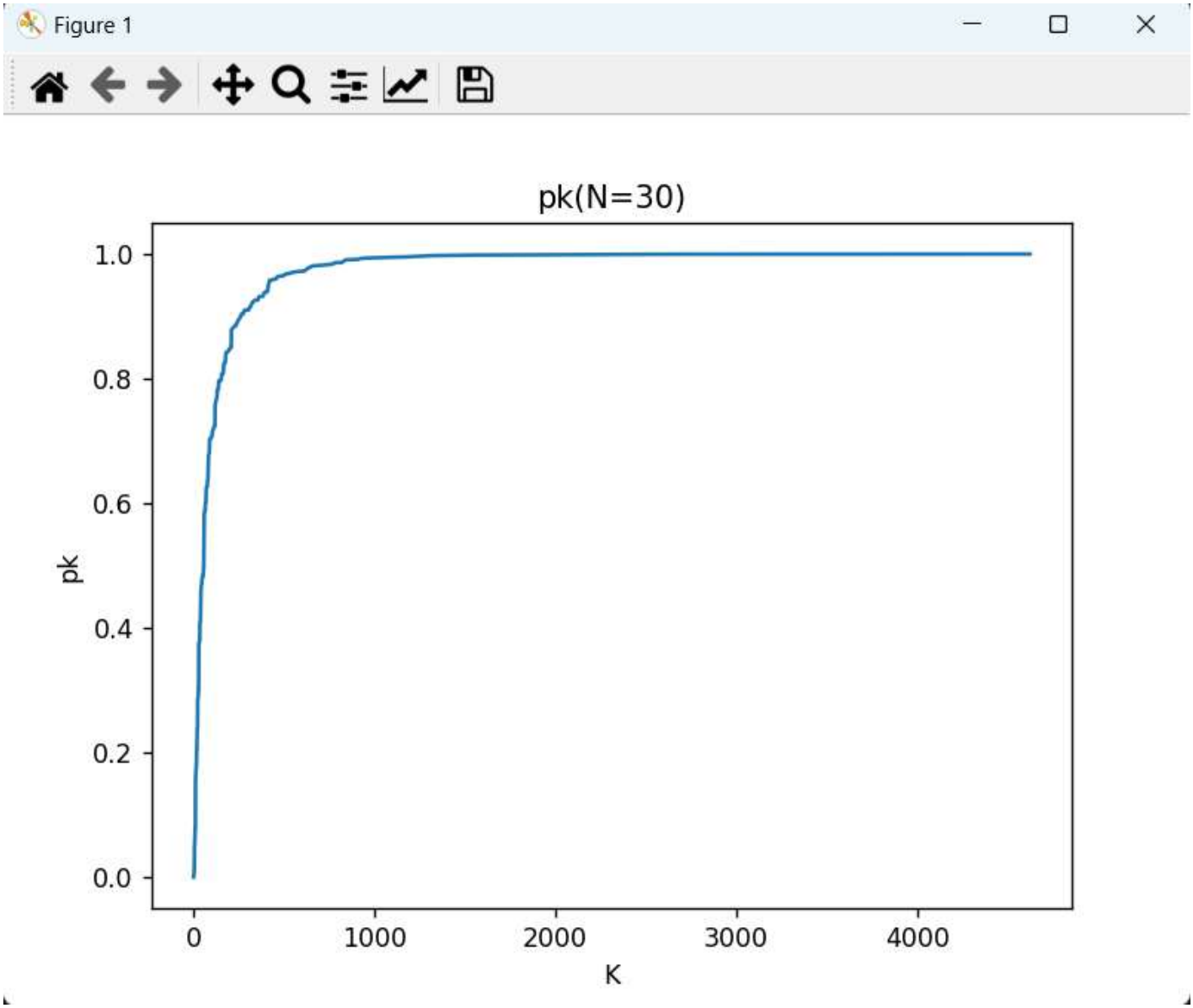
由用户输入`mu`与`N`，然后利用随机数得到`x_0`，选择随机取`N**2`个不同的`x_0`得到图像

实验测试

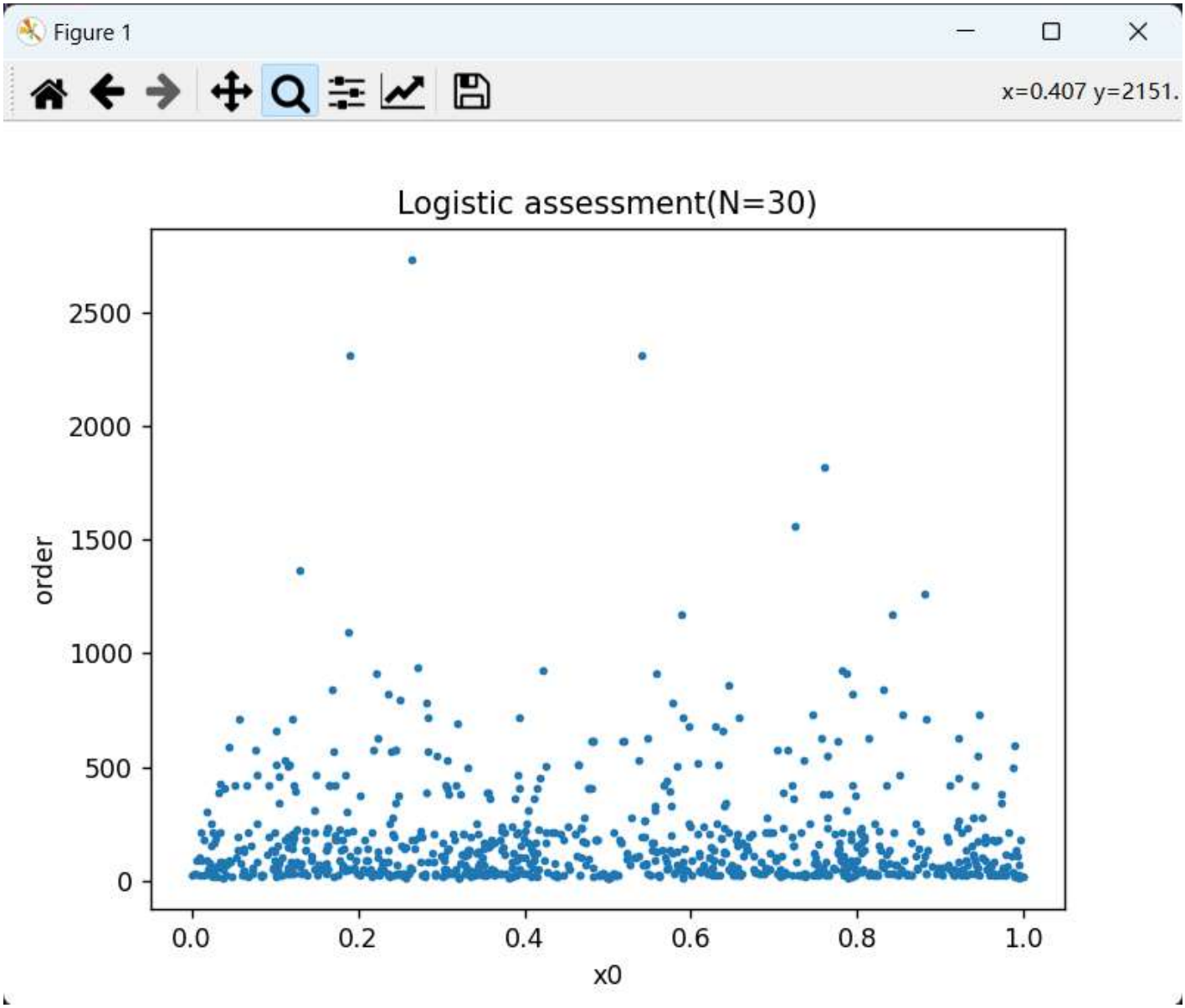
求N的最大阶

```
PS C:\Users\kyc\Desktop\密码学作业> & D:/python/python.exe c:/Users/kyc/Desktop/密码学作业/作业二/main.py
N=30时,绘制pk曲线所用时间为 1.6017985343933105
30个数的置乱最大阶为 4620
PS C:\Users\kyc\Desktop\密码学作业> & D:/python/python.exe c:/Users/kyc/Desktop/密码学作业/作业二/main.py
N=50时,绘制pk曲线所用时间为 2.385796546936035
50个数的置乱最大阶为 180180
PS C:\Users\kyc\Desktop\密码学作业> & D:/python/python.exe c:/Users/kyc/Desktop/密码学作业/作业二/main.py
N=80时,绘制pk曲线所用时间为 137.42185974121094
80个数的置乱最大阶为 19399380
PS C:\Users\kyc\Desktop\密码学作业>
```

绘制pk曲线



测评Logistic混沌映射



可以发现这样构造的时候，不管`x_0`初始值为多少，置乱分布比较均匀，但是大多数置乱的阶都比较低

实验总结

求最大阶的时候，我学会了善于利用现有资源~~我学会了不要重复造轮子~~

绘制pk曲线，复习了回溯算法（这里类似于深度有限搜索），还学会了使用`matplotlib`库

针对本次实验，不知道还有没有时间复杂度更低的算法，但是我已经尽力优化了我的代码

感谢陈嘉康同学告知我oeis.org这个网站，让我节省了很多时间