****

**J I A N G S U U N I V E R S I T Y**

《网络科学基础》

第二次平时作业



学院名称： 计算机学院

专业班级： 物联网2303班

学生姓名： 邱佳亮

学生学号： 3230611072

教师姓名： 熊书明

2024年10月

#### 题一代码

1. from math import log //导入相关包
2. import matplotlib.pyplot as plt
3. import numpy as np
4. *#%%*
5. c0=1
6. beta=3
7. k\_1=list(range(1,11))
8. k\_2=list(range(1,101))
9. y\_1=[]
10. y\_2=[]
11. for k\_val in k\_1:
12. y\_1.append(c0\*k\_val\*\*(-beta))
13. for k\_val in k\_2:
14. y\_2.append(c0\*k\_val\*\*(-beta))
15. *#%%*
16. fig,ax=plt.subplots(1,2,figsize=(8,6)) //创建子图
17. ax[0].plot(k\_1,y\_1,label='p(k)=c0\*k^(-beta)')
18. ax[1].plot(k\_2,y\_2,label='p(k)=c0\*k^(-beta)')
19. ax[0].set\_ylabel('p(k)')
20. ax[0].set\_xlabel('k')
21. ax[1].set\_ylabel('p(k)')
22. ax[1].set\_xlabel('k')
23. ax[0].legend(loc='best')
24. ax[1].legend(loc='best')
25. ax[0].grid()
26. ax[1].grid()
27. plt.tight\_layout()
28. plt.show()
29. *#%%*
30. y\_1 = []
31. y\_2 = []
32. for k\_val in k\_1:
33. y\_1.append(log(c0 \* k\_val \*\* (-beta)))
34. for k\_val in k\_2:
35. y\_2.append(log(c0 \* k\_val \*\* (-beta)))
36. fig, ax = plt.subplots(1, 2, figsize=(8, 6))
37. ax[0].plot(k\_1, y\_1, label='Ln(p(k)=c0\*k^(-beta))')
38. ax[1].plot(k\_2, y\_2, label='Ln(p(k)=c0\*k^(-beta))')
39. ax[0].set\_ylabel('Ln(p(k))')
40. ax[0].set\_xlabel('k')
41. ax[1].set\_ylabel('Ln(p(k))')
42. ax[1].set\_xlabel('k')
43. ax[0].legend(loc='best')
44. ax[1].legend(loc='best')
45. ax[0].grid()
46. ax[1].grid()
47. plt.show()

#### 运行结果

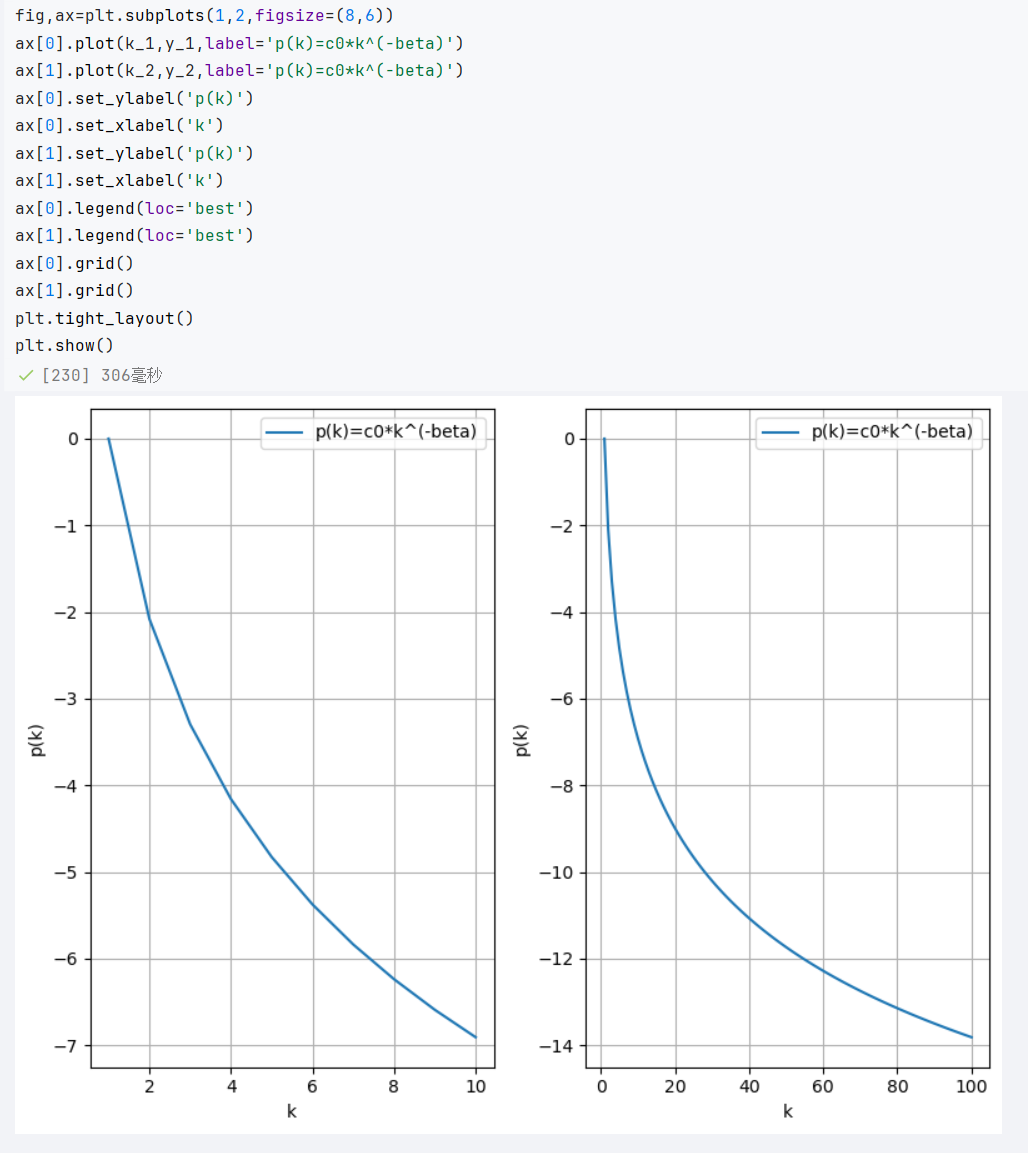


图 1 图形一图像

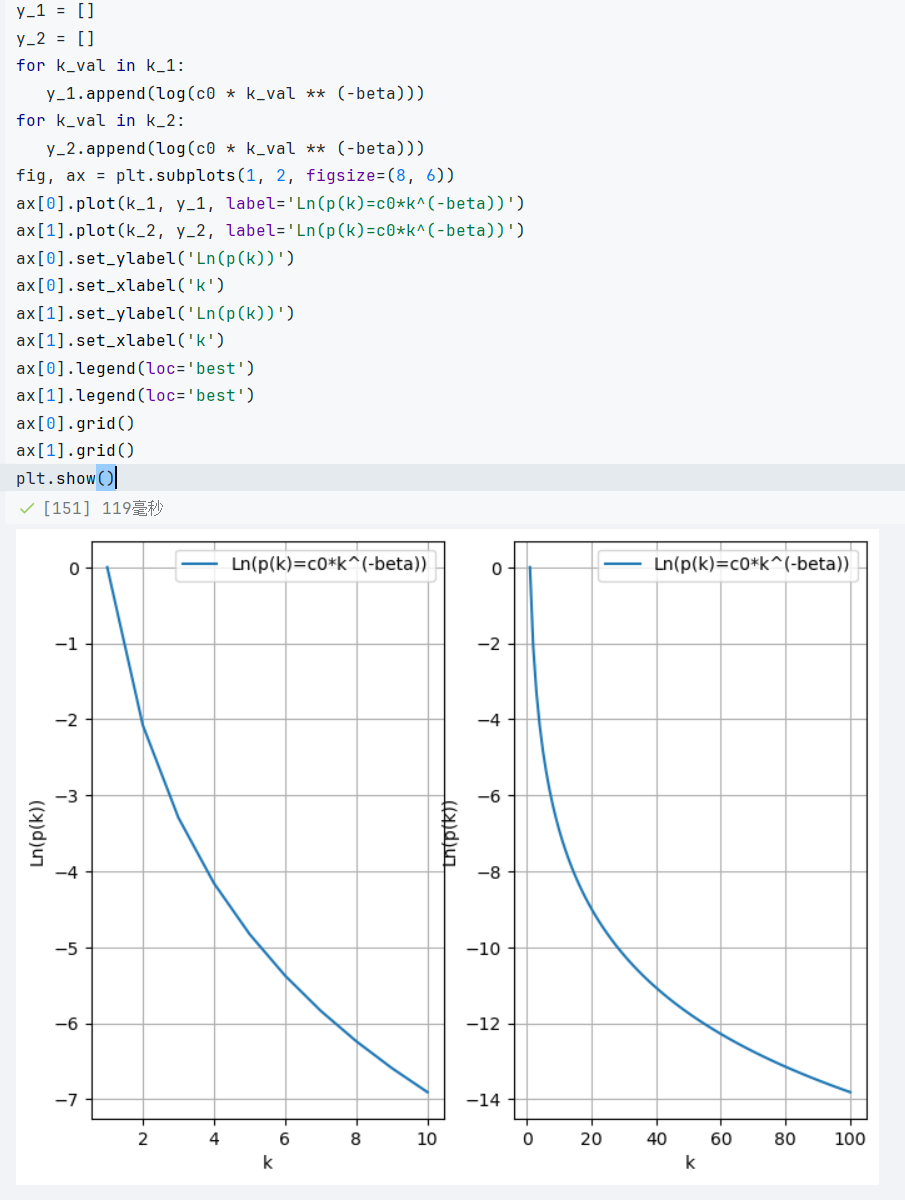


图 2 图形2图像

#### 题二代码

1. with open('martix.txt','r') as file: //读取包含节点邻接关系的文本
2. lines=file.readlines()
3. lines
4. *#%%*
5. import re //导入包
6. def extract(s):
7. pattern=r'\'[^\']\*\'' //正则表达式提取单引号中内容
8. res=re.findall(pattern,s)
9. res=list(map(lambda x:int(x.strip('\'')),res)) //剔除单引号 返回int列表
10. return res
11. martix=[]
12. for line in lines:
13. str=extract(line)
14. martix.append(str)
15. martix //二维数组
16. *#%%*
17. i=martix.\_\_len\_\_()
18. m=np.eye(i) //numpy创建单位矩阵
19. print(m)
20. *#%%*
21. for k,num in zip(martix,range(0,i)):
22. start=num
23. for t in k:
24. m[start][t-1]=1 //建立邻接表
25. print(m)
26. *#%%*
27. m\_=m
28. martix\_list=[m]
29. for i in range(1,11):
30. m\_=np.dot(m\_,m) //np矩阵相乘
31. filename=f"Martrix {i}.txt"
32. np.savetxt(filename,m\_,fmt='%-7d',delimiter='\t',newline='\n') //分别写入

#### 运行结果

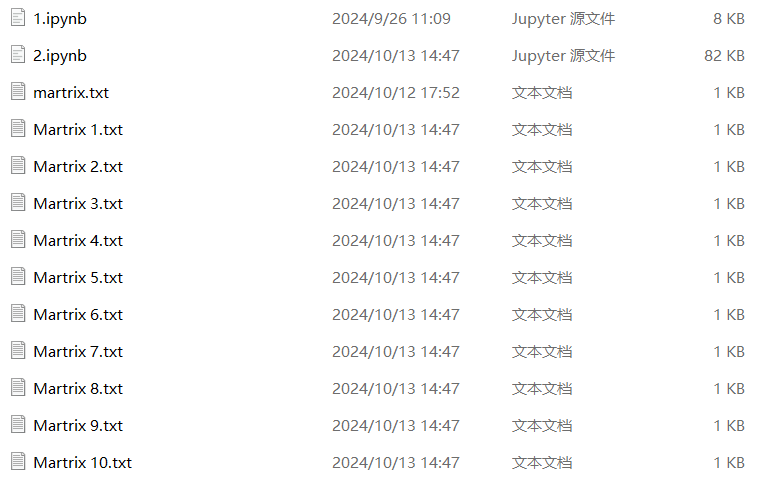


图 3 生成的邻接矩阵文件

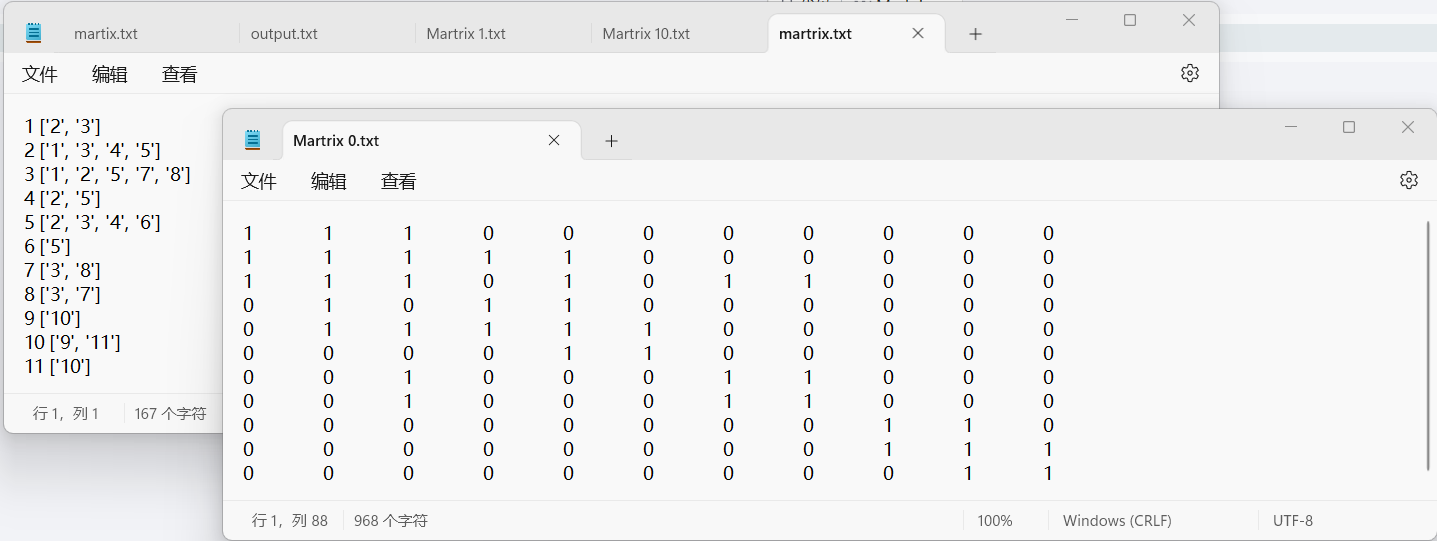


图 4 包含邻接矩阵的txt文件和节点关系文件