

# “网络科学基础”-第二次上机报告

班级： 物联网 2303                      姓名： 邱佳亮                      学号： 3230611072

上机日期： 2024.11.8，第十周周五下七八节课

2024 秋-网络科学基础（物联网 23）-第二次上机报告提交

## 一、上机题目

加强 Matlab 编程基础。

## 二、上机目的

1. 进一步熟悉 Matlab 编程上机环境、练习矩阵定位函数 find 的使用、练习子矩阵的赋值、删除等操作、练习常用函数的使用。

## 三、功能描述、上机程序（含必要的注释）、上机调试运行结果

2.1.1 使用方括号 “[ ]” 操作符产生一个列向量 x，内容为 1，2，4，7；使用方括号 “[ ]” 操作符产生一个行向量 x，内容为 1，2，4，7；使用冒号 “:” 操作符产生一个行向量 x，内容为 9，7，5，3，1；使用方括号 “[ ]” 操作符产生一个二维数组 A，第 1 行为 9，4，5，1；第 2 行为 1，0，4，7；使用 zeros 函数产生一个 3 \* 2 的二维数组 A，使用 ones 函数产生一个 3 \* 4 的二维数组 B，将 A、B 拼接成 3 \* 6 的二维数组 C；使用 rand 函数产生一个 3\*4 的二维数组 A，使用逻辑 1 标识的方式寻访 A 矩阵中大于 0.3 并且小于 0.7 的所有元素，赋给 B：

```

x=[1;2;4;7]

x = 4x1
     1
     2
     4
     7

x=[1 2 4 7]

x = 1x4
     1     2     4     7

x=[9:-2:1]

x = 1x5
     9     7     5     3     1

x=[9 4 5 1;1 0 4 7]

x = 2x4
     9     4     5     1
     1     0     4     7

A=zeros(3,2)

A = 3x2
     0     0
     0     0
     0     0

B=ones(3,4)

B = 3x4
     1     1     1     1
     1     1     1     1
     1     1     1     1

C=[A B]

C = 3x6
     0     0     1     1     1     1
     0     0     1     1     1     1
     0     0     1     1     1     1

A=rand(3,4)

A = 3x4
     0.8147     0.9134     0.2785     0.9649
     0.9058     0.6324     0.5469     0.1576
     0.1270     0.0975     0.9575     0.9706

B=A>0.3&A<0.7

B = 3x4 logical 数组
     0     0     0     0
     0     1     1     0
     0     0     0     0

```

图 1 运行结果

### 2.1.2 练习矩阵定位函数 find 的使用：

find 函数用于查找满足特定条件的元素。当使用 find(X >= 1) 时，如果没有指定输出参数，find 将返回满足条件的元素的线性索引。然而，当使用 [row, col, val]=find(X >= 1) 时，请求三个输出参数：满足条件的元素的行索引、列索引和

值。对于矩阵  $X$ ，`find(X >= 1)` 返回的是逻辑矩阵，其中满足条件  $X \geq 1$  的元素位置为 1（真），不满足条件的位置为 0（假）。当用三个输出参数调用 `find` 时，`val` 输出参数将返回原始矩阵  $X$  中满足条件的元素的值。

```
clear;
X=[3 2 0;-5 6 1]

X = 2x3
     3     2     0
    -5     6     1

indices=find(X>=1)

indices = 4x1
     1
     3
     4
     6

[row,col]=find(X>=1)

row = 4x1
     1
     1
     2
     2

col = 4x1
     1
     2
     2
     3

[row,col,val]=find(X>=1)

row = 4x1
     1
     1
     2
     2

col = 4x1
     1
     2
     2
     3

val = 4x1 logical 数组
     1
     1
     1
     1
```

图 2 运行结果

### 2.1.3 练习子矩阵的赋值、删除等操作

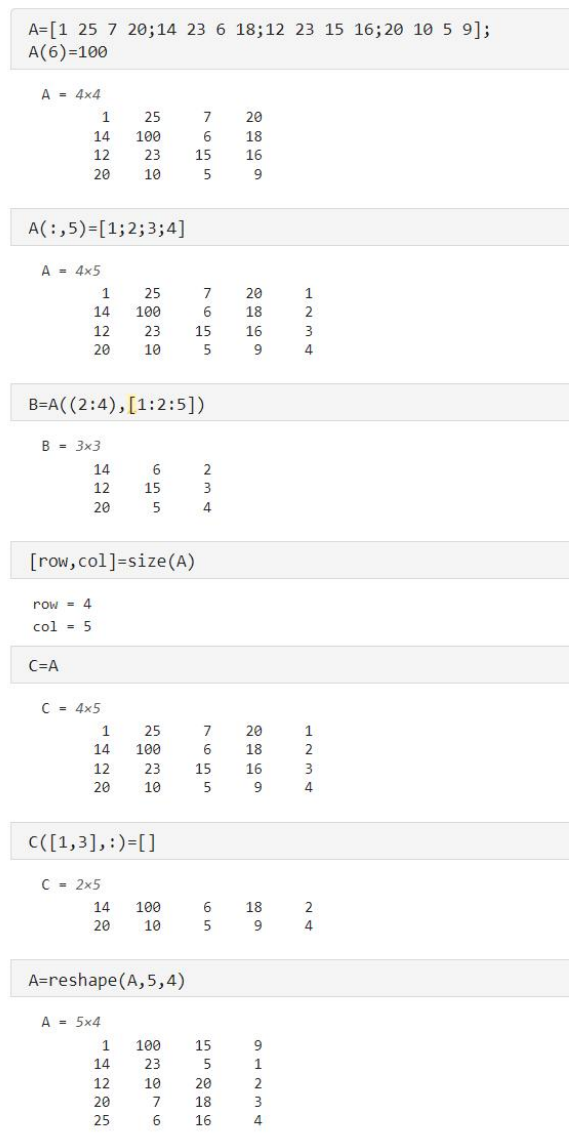


图 3 运行结果

## 2.1.4 练习常用函数的使用

```

x=[3 2 1 0]

x = 1x4
    3    2    1    0

a=length(x)

a = 4

b=size(x,1)

b = 1

c=size(x,2)

c = 4

s=sum(x)

s = 6

x=[3 2 1 0;5 6 8 7]

x = 2x4
    3    2    1    0
    5    6    8    7

b=size(x,1)

b = 2

c=size(x,2)

c = 4

d=size(x,3)

d = 1

s=sum(x)

s = 1x4
    8    8    9    7

s2=sum(x(:))

s2 = 32

```

图 4 运行结果

2.1.5 按要求编写 fun 函数：函数输入参数：一个向量；函数输出参数：该向量中所有大于 0 的元素的和；功能：求该向量中所有大于 0 的元素的和：

```

function s=fun(vect)
    vect1=vect(vect>0)
    s=sum(vect1)
end
s=fun([-1, 0, 2, 3, -5, 4])

vect1 = 1x3
    2    3    4

s = 9
s = 9

```

图 5 运行结果

2.2 在 Matlab 环境下，单步执行上机 1 中的程序代码

初始化一个  $n \times n$  的邻接矩阵  $a$ ，所有元素为 0，3.将顶点 1 与顶点 2、3、5 相连：

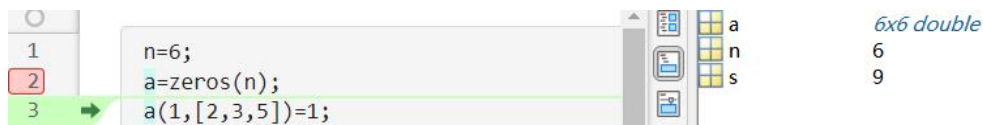


图 6 单步调试 1

将邻接矩阵  $a$  与其转置相加，得到无向图的邻接矩阵：

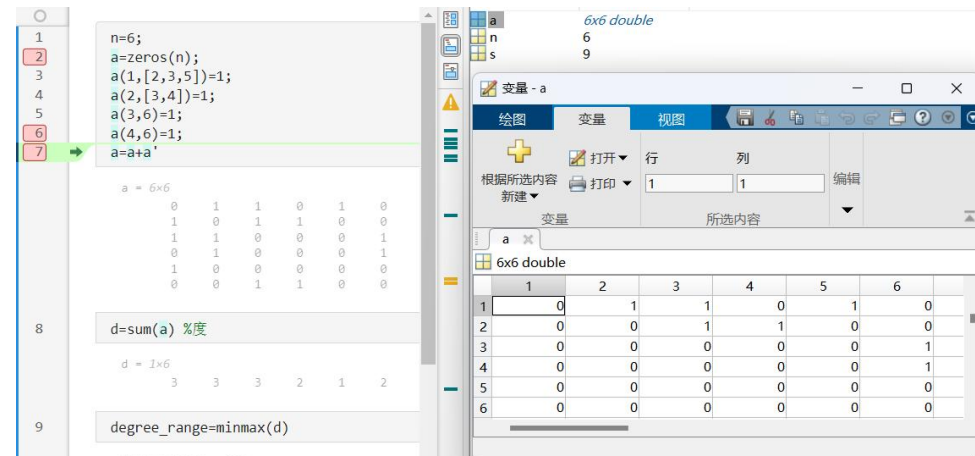


图 7 单步调试 2

计算每个顶点的度（与该顶点相连的边数），获取度的范围（最小值和最大值）：

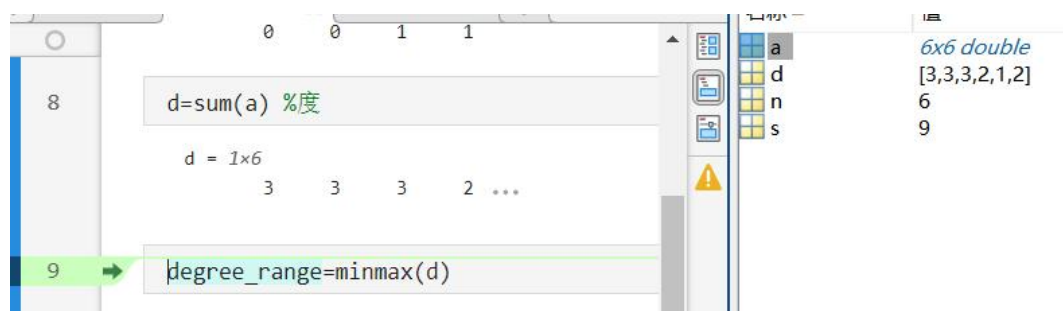


图 8 单步调试 3

创建一个包含所有可能度的向量，计算每个度的频数：

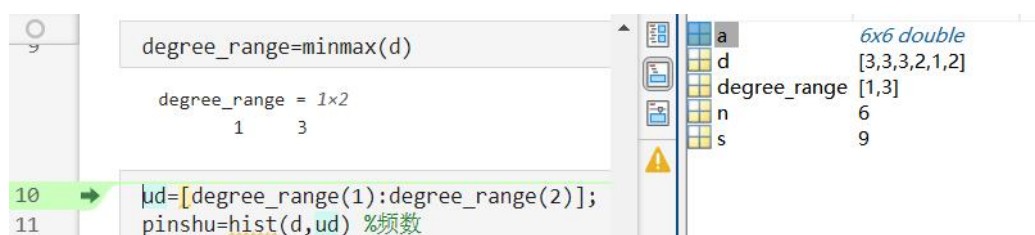


图 9 单步调试 4

创建一个频率分布表，其中第二列是每个度的频率（频数除以顶点数）：



图 10 单步调试 5

计算平均度:



图 11 单步调试 6

计算总边数:



图 12 单步调试 7

获取起点和终点的度，将起点和终点的度组合在一起，并按列排序:

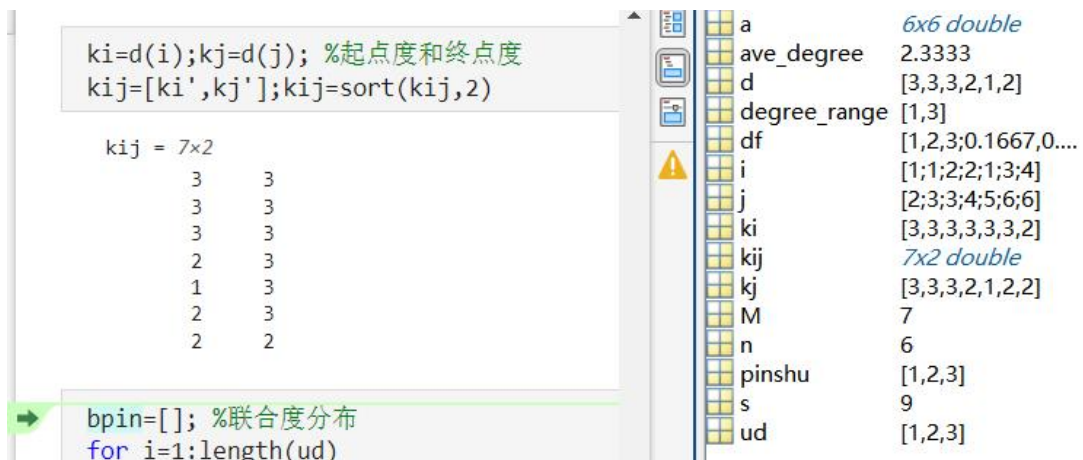


图 13 单步调试 8

初始化联合度分布的向量: 计算度为  $i$  和  $j$  的顶点对的数量, 如果找到度为

i 和 j 的顶点对，则记录其索引，将度为 i 和 j 的顶点对的频率添加到 bpin 中：

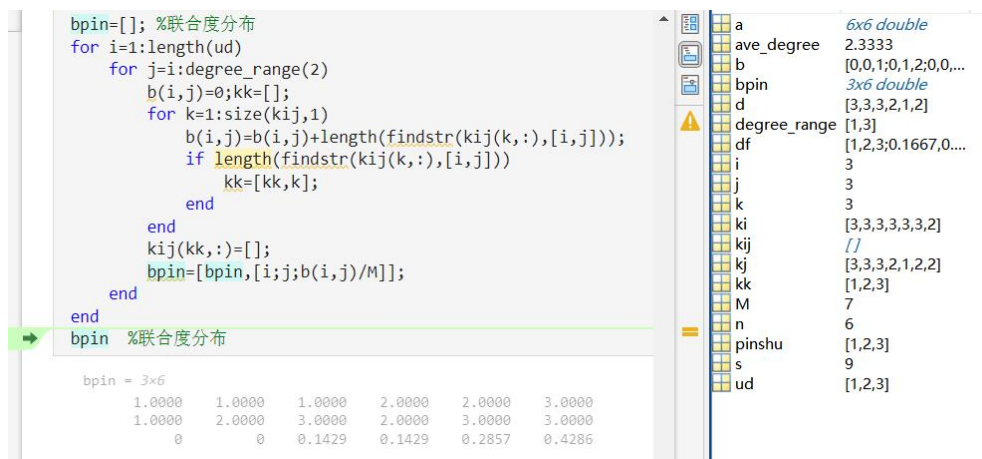


图 14 单步调试 9

计算最临近平均值：

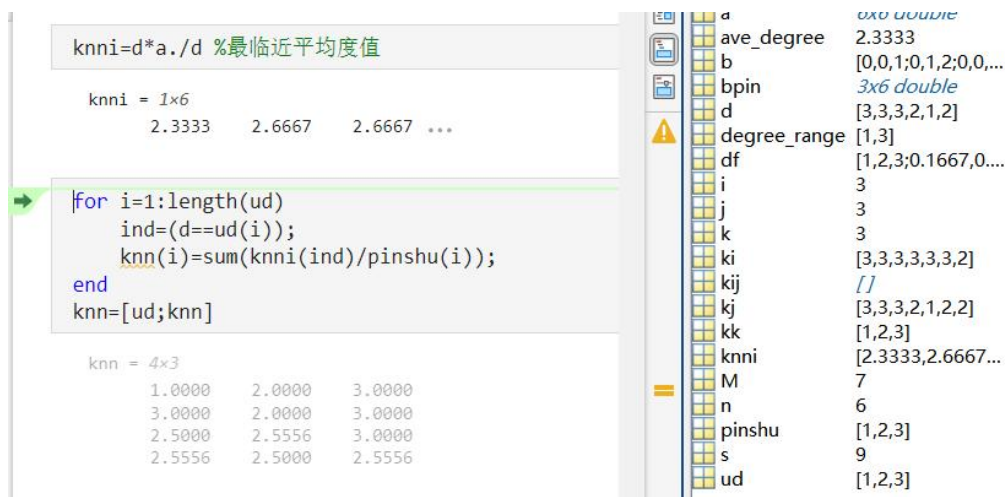


图 15 单步调试 10

计算每个度的 knn 值，创建 knn 值的表：

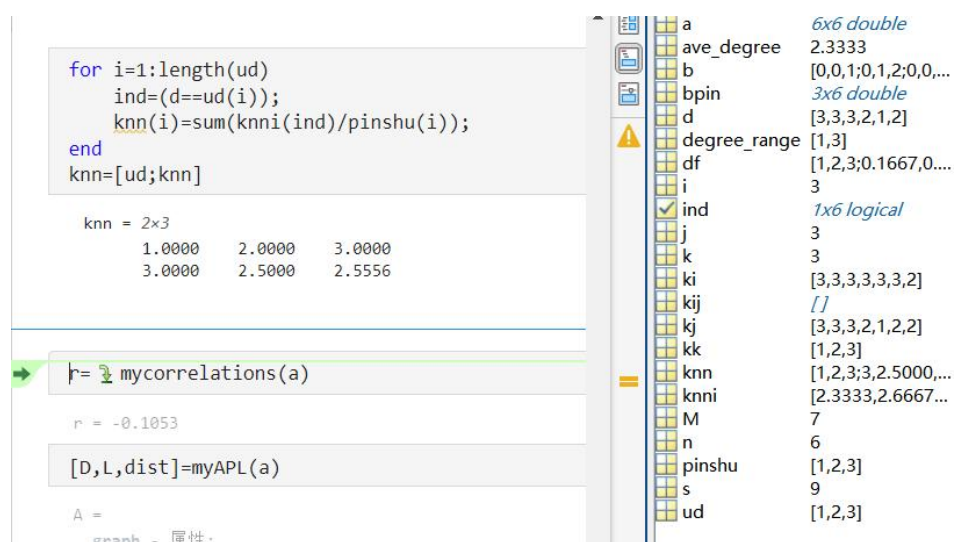


图 16 单步调试 11



计算并返回相关系数，计算直径，计算平均路径长度：

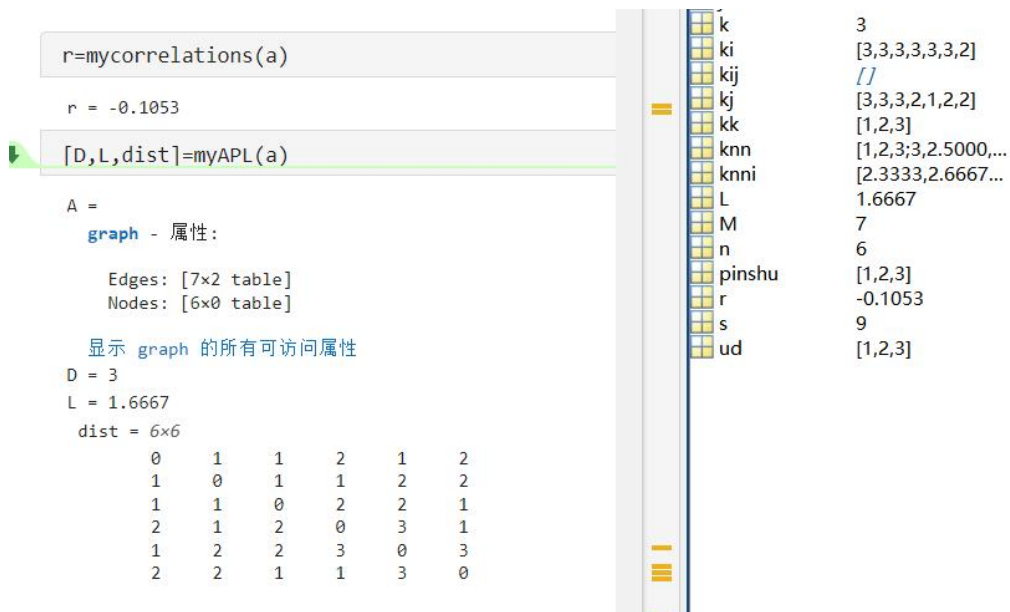


图 17 单步调试 12

#### 四、上机总结及感想

在本次“网络科学基础”的上机实验中，通过实际操作加深了对 Matlab 编程基础的理解。通过一系列的编程练习，不仅熟悉了 Matlab 的编程环境，还练习了矩阵定位函数 `find` 的使用、子矩阵的赋值与删除操作，以及常用函数的应用，这些技能对于理解和分析网络结构至关重要。实验过程中，我通过单步执行代码，逐步理解了邻接矩阵的构建、顶点度的计算以及网络特性的分析，这些操作让我对网络科学有了更直观的认识。这次实验不仅提升了我的编程技能，也加深了我对网络科学的兴趣，我期待在未来的学习中继续探索和进步。