HW 1 Report

Task 1

In [ ]:

```
! conda info
```

```
active environment : base
    active env location : /Users/dingzheyu/anaconda3
            shell level : 1
       user config file : /Users/dingzheyu/.condarc
 populated config files : /Users/dingzheyu/.condarc
          conda version : 4.7.11
    conda-build version : 3.10.5
         python version : 3.6.5.final.0
       virtual packages :
       base environment : /Users/dingzheyu/anaconda3  (writable)
           channel URLs : https://repo.anaconda.com/pkgs/main/osx-64
                          https://repo.anaconda.com/pkgs/main/noarch
                          https://repo.anaconda.com/pkgs/r/osx-64
                          https://repo.anaconda.com/pkgs/r/noarch
          package cache : /Users/dingzheyu/anaconda3/pkgs
                          /Users/dingzheyu/.conda/pkgs
       envs directories : /Users/dingzheyu/anaconda3/envs
                          /Users/dingzheyu/.conda/envs
               platform : osx-64
             user-agent : conda/4.7.11 requests/2.18.4 CPython/3.6.5 Darwin/18.6.0
OSX/10.14.5
                UID:GID : 501:20
             netrc file : None
           offline mode : False
```

Task 2

In [1]:

```
import numpy as np
import scipy.linalg
```

In [2]:

```
a = np.random.randint(1, 10, (10, 10))
```

In [3]:

```
np.ndim(a)
```

Out[3]:

```
2
```

In [4]:

```
np.size(a)
```

Out[4]:

```
100
```

In [5]:

```
np.shape(a)
```

Out[5]:

```
(10, 10)
```

In [6]:

```
a.shape[1]
```

Out[6]:

```
10
```

In [7]:

```
np.array([[1.,2.,3.], [4.,5.,6.]])
```

Out[7]:

```
array([[1., 2., 3.],
       [4., 5., 6.]])
```

In [8]:

```
b = np.random.randint(1, 10, (10, 10))
c = np.random.randint(1, 10, (10, 10))
d = np.random.randint(1, 10, (10, 10))
np.block([[a,b], [c,d]])
```

Out[8]:

```
array([[3, 2, 9, 6, 1, 6, 1, 4, 5, 8, 2, 6, 2, 9, 6, 7, 4, 1, 5, 7],
       [8, 5, 9, 3, 3, 5, 1, 9, 6, 8, 8, 9, 2, 5, 5, 2, 3, 4, 2, 1],
       [2, 6, 9, 3, 1, 4, 1, 1, 7, 7, 6, 5, 1, 5, 9, 3, 3, 2, 9, 6],
       [6, 7, 2, 5, 4, 8, 4, 5, 9, 9, 2, 3, 7, 8, 8, 4, 7, 9, 8, 3],
       [3, 5, 7, 8, 7, 8, 4, 6, 9, 8, 5, 2, 5, 4, 2, 8, 8, 8, 3, 7],
       [3, 8, 7, 9, 5, 4, 9, 2, 4, 8, 2, 5, 9, 2, 7, 9, 3, 7, 7, 7],
       [9, 6, 5, 2, 8, 4, 2, 7, 6, 6, 7, 4, 4, 6, 9, 9, 1, 7, 5, 2],
       [9, 1, 6, 3, 1, 4, 3, 1, 8, 4, 2, 4, 3, 6, 5, 1, 7, 2, 1, 7],
       [6, 1, 8, 9, 2, 6, 1, 7, 7, 6, 3, 5, 1, 9, 7, 8, 8, 5, 7, 9],
       [9, 4, 3, 2, 6, 2, 2, 6, 2, 8, 6, 4, 6, 3, 4, 2, 3, 4, 9, 3],
       [9, 5, 6, 3, 3, 6, 8, 8, 9, 1, 1, 2, 9, 9, 7, 7, 3, 6, 6, 4],
       [8, 9, 2, 5, 6, 3, 2, 8, 1, 4, 1, 4, 5, 8, 9, 1, 2, 6, 5, 4],
       [6, 9, 9, 7, 2, 6, 2, 2, 5, 3, 1, 2, 6, 6, 5, 2, 8, 6, 9, 4],
       [2, 9, 8, 9, 1, 1, 8, 6, 6, 9, 2, 5, 7, 3, 1, 5, 4, 1, 4, 9],
       [3, 4, 7, 2, 7, 3, 4, 5, 2, 7, 7, 2, 4, 6, 8, 9, 2, 4, 4, 8],
       [8, 3, 8, 8, 3, 4, 8, 5, 8, 2, 3, 6, 3, 5, 1, 2, 1, 2, 1, 2],
       [7, 2, 3, 4, 3, 1, 3, 3, 5, 3, 7, 6, 3, 3, 1, 9, 7, 1, 4, 5],
       [6, 1, 4, 8, 9, 6, 2, 3, 4, 6, 6, 6, 8, 9, 4, 6, 3, 3, 7, 1],
       [8, 3, 5, 9, 5, 2, 9, 5, 5, 8, 2, 6, 8, 4, 7, 3, 2, 2, 2, 1],
       [3, 4, 9, 7, 9, 5, 4, 2, 9, 8, 7, 3, 9, 7, 2, 8, 3, 5, 1, 2]])
```

In [9]:

```
a[-1]
```

Out[9]:

```
array([9, 4, 3, 2, 6, 2, 2, 6, 2, 8])
```

In [10]:

```
a[1,4]
```

Out[10]:

```
3
```

In [11]:

```
print(a[0:5])
print(a[:5])
print(a[0:5,:])
```

```
[[3 2 9 6 1 6 1 4 5 8]
 [8 5 9 3 3 5 1 9 6 8]
 [2 6 9 3 1 4 1 1 7 7]
 [6 7 2 5 4 8 4 5 9 9]
 [3 5 7 8 7 8 4 6 9 8]]
[[3 2 9 6 1 6 1 4 5 8]
 [8 5 9 3 3 5 1 9 6 8]
 [2 6 9 3 1 4 1 1 7 7]
 [6 7 2 5 4 8 4 5 9 9]
 [3 5 7 8 7 8 4 6 9 8]]
[[3 2 9 6 1 6 1 4 5 8]
 [8 5 9 3 3 5 1 9 6 8]
 [2 6 9 3 1 4 1 1 7 7]
 [6 7 2 5 4 8 4 5 9 9]
 [3 5 7 8 7 8 4 6 9 8]]
```

In [12]:

```
a[-5:]
```

Out[12]:

```
array([[3, 8, 7, 9, 5, 4, 9, 2, 4, 8],
       [9, 6, 5, 2, 8, 4, 2, 7, 6, 6],
       [9, 1, 6, 3, 1, 4, 3, 1, 8, 4],
       [6, 1, 8, 9, 2, 6, 1, 7, 7, 6],
       [9, 4, 3, 2, 6, 2, 2, 6, 2, 8]])
```

In [13]:

```
a[0:3][:,4:9]
```

Out[13]:

```
array([[1, 6, 1, 4, 5],
       [3, 5, 1, 9, 6],
       [1, 4, 1, 1, 7]])
```

In [14]:

```
a[([1,3],[0,2])]
```

Out[14]:

```
array([8, 2])
```

In [15]:

```
a[ 2:1:2,:]
```

Out[15]:

```
array([], shape=(0, 10), dtype=int64)
```

In [16]:

```
a[ ::2,:]
```

Out[16]:

```
array([[3, 2, 9, 6, 1, 6, 1, 4, 5, 8],
       [2, 6, 9, 3, 1, 4, 1, 1, 7, 7],
       [3, 5, 7, 8, 7, 8, 4, 6, 9, 8],
       [9, 6, 5, 2, 8, 4, 2, 7, 6, 6],
       [6, 1, 8, 9, 2, 6, 1, 7, 7, 6]])
```

In [17]:

```
a[ ::-1,:]
```

Out[17]:

```
array([[9, 4, 3, 2, 6, 2, 2, 6, 2, 8],
       [6, 1, 8, 9, 2, 6, 1, 7, 7, 6],
       [9, 1, 6, 3, 1, 4, 3, 1, 8, 4],
       [9, 6, 5, 2, 8, 4, 2, 7, 6, 6],
       [3, 8, 7, 9, 5, 4, 9, 2, 4, 8],
       [3, 5, 7, 8, 7, 8, 4, 6, 9, 8],
       [6, 7, 2, 5, 4, 8, 4, 5, 9, 9],
       [2, 6, 9, 3, 1, 4, 1, 1, 7, 7],
       [8, 5, 9, 3, 3, 5, 1, 9, 6, 8],
       [3, 2, 9, 6, 1, 6, 1, 4, 5, 8]])
```

In [18]:

```
a[:len(a),0]
```

Out[18]:

```
array([3, 8, 2, 6, 3, 3, 9, 9, 6, 9])
```

In [19]:

```
a.transpose()
```

Out[19]:

```
array([[3, 8, 2, 6, 3, 3, 9, 9, 6, 9],
       [2, 5, 6, 7, 5, 8, 6, 1, 1, 4],
       [9, 9, 9, 2, 7, 7, 5, 6, 8, 3],
       [6, 3, 3, 5, 8, 9, 2, 3, 9, 2],
       [1, 3, 1, 4, 7, 5, 8, 1, 2, 6],
       [6, 5, 4, 8, 8, 4, 4, 4, 6, 2],
       [1, 1, 1, 4, 4, 9, 2, 3, 1, 2],
       [4, 9, 1, 5, 6, 2, 7, 1, 7, 6],
       [5, 6, 7, 9, 9, 4, 6, 8, 7, 2],
       [8, 8, 7, 9, 8, 8, 6, 4, 6, 8]])
```

In [20]:

```
a.conj().transpose()
```

Out[20]:

```
array([[3, 8, 2, 6, 3, 3, 9, 9, 6, 9],
       [2, 5, 6, 7, 5, 8, 6, 1, 1, 4],
       [9, 9, 9, 2, 7, 7, 5, 6, 8, 3],
       [6, 3, 3, 5, 8, 9, 2, 3, 9, 2],
       [1, 3, 1, 4, 7, 5, 8, 1, 2, 6],
       [6, 5, 4, 8, 8, 4, 4, 4, 6, 2],
       [1, 1, 1, 4, 4, 9, 2, 3, 1, 2],
       [4, 9, 1, 5, 6, 2, 7, 1, 7, 6],
       [5, 6, 7, 9, 9, 4, 6, 8, 7, 2],
       [8, 8, 7, 9, 8, 8, 6, 4, 6, 8]])
```

In [21]:

```
a @ b
```

Out[21]:

```
array([[183, 208, 189, 245, 297, 207, 206, 205, 309, 243],
       [232, 280, 201, 326, 347, 256, 270, 219, 327, 323],
       [197, 213, 143, 225, 268, 189, 179, 179, 276, 211],
       [245, 289, 249, 333, 353, 317, 280, 290, 339, 311],
       [270, 293, 277, 360, 401, 342, 326, 338, 390, 352],
       [290, 278, 247, 328, 382, 301, 251, 317, 356, 253],
       [230, 265, 196, 318, 308, 283, 276, 235, 283, 314],
       [152, 196, 135, 260, 271, 239, 184, 167, 264, 241],
       [186, 237, 217, 320, 348, 258, 272, 245, 339, 306],
       [186, 207, 167, 250, 237, 210, 209, 179, 230, 235]])
```

In [22]:

```
a * b
```

Out[22]:

```
array([[ 6, 12, 18, 54,  6, 42,  4,  4, 25, 56],
       [64, 45, 18, 15, 15, 10,  3, 36, 12,  8],
       [12, 30,  9, 15,  9, 12,  3,  2, 63, 42],
       [12, 21, 14, 40, 32, 32, 28, 45, 72, 27],
       [15, 10, 35, 32, 14, 64, 32, 48, 27, 56],
       [ 6, 40, 63, 18, 35, 36, 27, 14, 28, 56],
       [63, 24, 20, 12, 72, 36,  2, 49, 30, 12],
       [18,  4, 18, 18,  5,  4, 21,  2,  8, 28],
       [18,  5,  8, 81, 14, 48,  8, 35, 49, 54],
       [54, 16, 18,  6, 24,  4,  6, 24, 18, 24]])
```

In [23]:

```
a / b
```

Out[23]:

```
array([[1.5       , 0.33333333, 4.5       , 0.66666667, 0.16666667,
        0.85714286, 0.25      , 4.        , 1.        , 1.14285714],
       [1.        , 0.55555556, 4.5       , 0.6       , 0.6       ,
        2.5       , 0.33333333, 2.25      , 3.        , 8.        ],
       [0.33333333, 1.2       , 9.        , 0.6       , 0.11111111,
        1.33333333, 0.33333333, 0.5       , 0.77777778, 1.16666667],
       [3.        , 2.33333333, 0.28571429, 0.625     , 0.5       ,
        2.        , 0.57142857, 0.55555556, 1.125     , 3.        ],
       [0.6       , 2.5       , 1.4       , 2.        , 3.5       ,
        1.        , 0.5       , 0.75      , 3.        , 1.14285714],
       [1.5       , 1.6       , 0.77777778, 4.5       , 0.71428571,
        0.44444444, 3.        , 0.28571429, 0.57142857, 1.14285714],
       [1.28571429, 1.5       , 1.25      , 0.33333333, 0.88888889,
        0.44444444, 2.        , 1.        , 1.2       , 3.        ],
       [4.5       , 0.25      , 2.        , 0.5       , 0.2       ,
        4.        , 0.42857143, 0.5       , 8.        , 0.57142857],
       [2.        , 0.2       , 8.        , 1.        , 0.28571429,
        0.75      , 0.125     , 1.4       , 1.        , 0.66666667],
       [1.5       , 1.        , 0.5       , 0.66666667, 1.5       ,
        1.        , 0.66666667, 1.5       , 0.22222222, 2.66666667]])
```

In [24]:

```
a ** 3
```

Out[24]:

```
array([[ 27,   8, 729, 216,   1, 216,   1,  64, 125, 512],
       [512, 125, 729,  27,  27, 125,   1, 729, 216, 512],
       [  8, 216, 729,  27,   1,  64,   1,   1, 343, 343],
       [216, 343,   8, 125,  64, 512,  64, 125, 729, 729],
       [ 27, 125, 343, 512, 343, 512,  64, 216, 729, 512],
       [ 27, 512, 343, 729, 125,  64, 729,   8,  64, 512],
       [729, 216, 125,   8, 512,  64,   8, 343, 216, 216],
       [729,   1, 216,  27,   1,  64,  27,   1, 512,  64],
       [216,   1, 512, 729,   8, 216,   1, 343, 343, 216],
       [729,  64,  27,   8, 216,   8,   8, 216,   8, 512]])
```

In [25]:

```
a > 0.5
```

Out[25]:

```
array([[ True,  True,  True,  True,  True,  True,  True,  True,  True,
         True],
       [ True,  True,  True,  True,  True,  True,  True,  True,  True,
         True],
       [ True,  True,  True,  True,  True,  True,  True,  True,  True,
         True],
       [ True,  True,  True,  True,  True,  True,  True,  True,  True,
         True],
       [ True,  True,  True,  True,  True,  True,  True,  True,  True,
         True],
       [ True,  True,  True,  True,  True,  True,  True,  True,  True,
         True],
       [ True,  True,  True,  True,  True,  True,  True,  True,  True,
         True],
       [ True,  True,  True,  True,  True,  True,  True,  True,  True,
         True],
       [ True,  True,  True,  True,  True,  True,  True,  True,  True,
         True],
       [ True,  True,  True,  True,  True,  True,  True,  True,  True,
         True]])
```

In [26]:

```
np.nonzero(a>0.5)
```

Out[26]:

```
(array([0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2,
2,
        2, 2, 2, 2, 2, 2, 2, 2, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 4, 4, 4,
4,
        4, 4, 4, 4, 4, 4, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 6, 6, 6, 6, 6,
6,
        6, 6, 6, 6, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 8, 8, 8, 8, 8, 8, 8,
8,
        8, 8, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9]),
 array([0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 0,
1,
        2, 3, 4, 5, 6, 7, 8, 9, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 0, 1, 2,
3,
        4, 5, 6, 7, 8, 9, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 0, 1, 2, 3, 4,
5,
        6, 7, 8, 9, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 0, 1, 2, 3, 4, 5, 6,
7,
        8, 9, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9]))
```

In [27]:

```
v = np.random.randint(-2, 10, 10)
a[:,np.nonzero(v>0.5)[0]]
```

Out[27]:

```
array([[3, 2, 1, 1, 4, 8],
       [8, 5, 3, 1, 9, 8],
       [2, 6, 1, 1, 1, 7],
       [6, 7, 4, 4, 5, 9],
       [3, 5, 7, 4, 6, 8],
       [3, 8, 5, 9, 2, 8],
       [9, 6, 8, 2, 7, 6],
       [9, 1, 1, 3, 1, 4],
       [6, 1, 2, 1, 7, 6],
       [9, 4, 6, 2, 6, 8]])
```

In [28]:

```
a[:,v.T>0.5]
```

Out[28]:

```
array([[3, 2, 1, 1, 4, 8],
       [8, 5, 3, 1, 9, 8],
       [2, 6, 1, 1, 1, 7],
       [6, 7, 4, 4, 5, 9],
       [3, 5, 7, 4, 6, 8],
       [3, 8, 5, 9, 2, 8],
       [9, 6, 8, 2, 7, 6],
       [9, 1, 1, 3, 1, 4],
       [6, 1, 2, 1, 7, 6],
       [9, 4, 6, 2, 6, 8]])
```

In [29]:

```
a[a<0.5]=0
```

In [30]:

```
a * (a>0.5)
```

Out[30]:

```
array([[3, 2, 9, 6, 1, 6, 1, 4, 5, 8],
       [8, 5, 9, 3, 3, 5, 1, 9, 6, 8],
       [2, 6, 9, 3, 1, 4, 1, 1, 7, 7],
       [6, 7, 2, 5, 4, 8, 4, 5, 9, 9],
       [3, 5, 7, 8, 7, 8, 4, 6, 9, 8],
       [3, 8, 7, 9, 5, 4, 9, 2, 4, 8],
       [9, 6, 5, 2, 8, 4, 2, 7, 6, 6],
       [9, 1, 6, 3, 1, 4, 3, 1, 8, 4],
       [6, 1, 8, 9, 2, 6, 1, 7, 7, 6],
       [9, 4, 3, 2, 6, 2, 2, 6, 2, 8]])
```

In [31]:

```
a[:] = 3
```

In [32]:

```
y = a.copy()
```

In [33]:

```
y = a[1,:].copy()
```

In [34]:

```
y = a.flatten()
```

In [35]:

```
np.arange(1.,11.)
```

Out[35]:

```
array([ 1.,  2.,  3.,  4.,  5.,  6.,  7.,  8.,  9., 10.])
```

In [36]:

```
np.arange(10.)
```

Out[36]:

```
array([0., 1., 2., 3., 4., 5., 6., 7., 8., 9.])
```

In [37]:

```
np.arange(1.,11.)[:]
```

Out[37]:

```
array([ 1.,  2.,  3.,  4.,  5.,  6.,  7.,  8.,  9., 10.])
```

In [38]:

```
np.zeros((3,4))
```

Out[38]:

```
array([[0., 0., 0., 0.],
       [0., 0., 0., 0.],
       [0., 0., 0., 0.]])
```

In [39]:

```
np.ones((3,4))
```

Out[39]:

```
array([[1., 1., 1., 1.],
       [1., 1., 1., 1.],
       [1., 1., 1., 1.]])
```

In [40]:

```
np.eye(3)
```

Out[40]:

```
array([[1., 0., 0.],
       [0., 1., 0.],
       [0., 0., 1.]])
```

In [41]:

```
np.diag(a)
```

Out[41]:

```
array([3, 3, 3, 3, 3, 3, 3, 3, 3, 3])
```

In [42]:

```
np.diag(a,0)
```

Out[42]:

```
array([3, 3, 3, 3, 3, 3, 3, 3, 3, 3])
```

In [43]:

```
np.random.rand(3,4)
```

Out[43]:

```
array([[0.30684848, 0.98428237, 0.76974404, 0.95529201],
       [0.94798779, 0.64498183, 0.61244636, 0.92778866],
       [0.69491876, 0.89019147, 0.18871753, 0.09431126]])
```

In [44]:

```
np.linspace(1,3,4)
```

Out[44]:

```
array([1.        , 1.66666667, 2.33333333, 3.        ])
```

In [45]:

```
np.mgrid[0:9.,0:6.]
```

Out[45]:

```
array([[[0., 0., 0., 0., 0., 0.],
        [1., 1., 1., 1., 1., 1.],
        [2., 2., 2., 2., 2., 2.],
        [3., 3., 3., 3., 3., 3.],
        [4., 4., 4., 4., 4., 4.],
        [5., 5., 5., 5., 5., 5.],
        [6., 6., 6., 6., 6., 6.],
        [7., 7., 7., 7., 7., 7.],
        [8., 8., 8., 8., 8., 8.]],

       [[0., 1., 2., 3., 4., 5.],
        [0., 1., 2., 3., 4., 5.],
        [0., 1., 2., 3., 4., 5.],
        [0., 1., 2., 3., 4., 5.],
        [0., 1., 2., 3., 4., 5.],
        [0., 1., 2., 3., 4., 5.],
        [0., 1., 2., 3., 4., 5.],
        [0., 1., 2., 3., 4., 5.],
        [0., 1., 2., 3., 4., 5.]]])
```

In [46]:

```
np.ogrid[0:9.,0:6.]
```

Out[46]:

```
[array([[0.],
        [1.],
        [2.],
        [3.],
        [4.],
        [5.],
        [6.],
        [7.],
        [8.]]), array([[0., 1., 2., 3., 4., 5.]])]
```

In [47]:

```
np.meshgrid([1,2,4],[2,4,5])
```

Out[47]:

```
[array([[1, 2, 4],
        [1, 2, 4],
        [1, 2, 4]]), array([[2, 2, 2],
        [4, 4, 4],
        [5, 5, 5]])]
```

In [48]:

```
np.tile(a, (100, 1))
```

Out[48]:

```
array([[3, 3, 3, ..., 3, 3, 3],
       [3, 3, 3, ..., 3, 3, 3],
       [3, 3, 3, ..., 3, 3, 3],
       ...,
       [3, 3, 3, ..., 3, 3, 3],
       [3, 3, 3, ..., 3, 3, 3],
       [3, 3, 3, ..., 3, 3, 3]])
```

In [49]:

```
np.concatenate((a,b),1)
np.hstack((a,b))
np.column_stack((a,b))
```

Out[49]:

```
array([[3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 2, 6, 2, 9, 6, 7, 4, 1, 5, 7],
       [3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 8, 9, 2, 5, 5, 2, 3, 4, 2, 1],
       [3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 6, 5, 1, 5, 9, 3, 3, 2, 9, 6],
       [3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 2, 3, 7, 8, 8, 4, 7, 9, 8, 3],
       [3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 5, 2, 5, 4, 2, 8, 8, 8, 3, 7],
       [3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 2, 5, 9, 2, 7, 9, 3, 7, 7, 7],
       [3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 7, 4, 4, 6, 9, 9, 1, 7, 5, 2],
       [3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 2, 4, 3, 6, 5, 1, 7, 2, 1, 7],
       [3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 5, 1, 9, 7, 8, 8, 5, 7, 9],
       [3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 6, 4, 6, 3, 4, 2, 3, 4, 9, 3]])
```

In [50]:

```
a.max()
```

Out[50]:

```
3
```

In [51]:

```
a.max(0)
```

Out[51]:

```
array([3, 3, 3, 3, 3, 3, 3, 3, 3, 3])
```

In [52]:

```
np.maximum(a, b)
```

Out[52]:

```
array([[3, 6, 3, 9, 6, 7, 4, 3, 5, 7],
       [8, 9, 3, 5, 5, 3, 3, 4, 3, 3],
       [6, 5, 3, 5, 9, 3, 3, 3, 9, 6],
       [3, 3, 7, 8, 8, 4, 7, 9, 8, 3],
       [5, 3, 5, 4, 3, 8, 8, 8, 3, 7],
       [3, 5, 9, 3, 7, 9, 3, 7, 7, 7],
       [7, 4, 4, 6, 9, 9, 3, 7, 5, 3],
       [3, 4, 3, 6, 5, 3, 7, 3, 3, 7],
       [3, 5, 3, 9, 7, 8, 8, 5, 7, 9],
       [6, 4, 6, 3, 4, 3, 3, 4, 9, 3]])
```

In [53]:

```
np.sqrt(v @ v)
```

Out[53]:

```
10.14889156509222
```

In [54]:

```
np.logical_and(a,b)
```

Out[54]:

```
array([[ True,  True,  True,  True,  True,  True,  True,  True,  True,
         True],
       [ True,  True,  True,  True,  True,  True,  True,  True,  True,
         True],
       [ True,  True,  True,  True,  True,  True,  True,  True,  True,
         True],
       [ True,  True,  True,  True,  True,  True,  True,  True,  True,
         True],
       [ True,  True,  True,  True,  True,  True,  True,  True,  True,
         True],
       [ True,  True,  True,  True,  True,  True,  True,  True,  True,
         True],
       [ True,  True,  True,  True,  True,  True,  True,  True,  True,
         True],
       [ True,  True,  True,  True,  True,  True,  True,  True,  True,
         True],
       [ True,  True,  True,  True,  True,  True,  True,  True,  True,
         True],
       [ True,  True,  True,  True,  True,  True,  True,  True,  True,
         True]])
```

In [55]:

```
a & b
```

Out[55]:

```
array([[2, 2, 2, 1, 2, 3, 0, 1, 1, 3],
       [0, 1, 2, 1, 1, 2, 3, 0, 2, 1],
       [2, 1, 1, 1, 1, 3, 3, 2, 1, 2],
       [2, 3, 3, 0, 0, 0, 3, 1, 0, 3],
       [1, 2, 1, 0, 2, 0, 0, 0, 3, 3],
       [2, 1, 1, 2, 3, 1, 3, 3, 3, 3],
       [3, 0, 0, 2, 1, 1, 1, 3, 1, 2],
       [2, 0, 3, 2, 1, 1, 3, 2, 1, 3],
       [3, 1, 1, 1, 3, 0, 0, 1, 3, 1],
       [2, 0, 2, 3, 0, 2, 3, 0, 1, 3]])
```

In [56]:

```
a | b
```

Out[56]:

```
array([[ 3,  7,  3, 11,  7,  7,  7,  3,  7,  7],
       [11, 11,  3,  7,  7,  3,  3,  7,  3,  3],
       [ 7,  7,  3,  7, 11,  3,  3,  3, 11,  7],
       [ 3,  3,  7, 11, 11,  7,  7, 11, 11,  3],
       [ 7,  3,  7,  7,  3, 11, 11, 11,  3,  7],
       [ 3,  7, 11,  3,  7, 11,  3,  7,  7,  7],
       [ 7,  7,  7,  7, 11, 11,  3,  7,  7,  3],
       [ 3,  7,  3,  7,  7,  3,  7,  3,  3,  7],
       [ 3,  7,  3, 11,  7, 11, 11,  7,  7, 11],
       [ 7,  7,  7,  3,  7,  3,  3,  7, 11,  3]])
```

In [57]:

```
np.linalg.inv(a[1:2, 1:2])
```

Out[57]:

```
array([[0.33333333]])
```

In [58]:

```
np.linalg.pinv(a)
```

Out[58]:

```
array([[0.00333333, 0.00333333, 0.00333333, 0.00333333, 0.00333333,
        0.00333333, 0.00333333, 0.00333333, 0.00333333, 0.00333333],
       [0.00333333, 0.00333333, 0.00333333, 0.00333333, 0.00333333,
        0.00333333, 0.00333333, 0.00333333, 0.00333333, 0.00333333],
       [0.00333333, 0.00333333, 0.00333333, 0.00333333, 0.00333333,
        0.00333333, 0.00333333, 0.00333333, 0.00333333, 0.00333333],
       [0.00333333, 0.00333333, 0.00333333, 0.00333333, 0.00333333,
        0.00333333, 0.00333333, 0.00333333, 0.00333333, 0.00333333],
       [0.00333333, 0.00333333, 0.00333333, 0.00333333, 0.00333333,
        0.00333333, 0.00333333, 0.00333333, 0.00333333, 0.00333333],
       [0.00333333, 0.00333333, 0.00333333, 0.00333333, 0.00333333,
        0.00333333, 0.00333333, 0.00333333, 0.00333333, 0.00333333],
       [0.00333333, 0.00333333, 0.00333333, 0.00333333, 0.00333333,
        0.00333333, 0.00333333, 0.00333333, 0.00333333, 0.00333333],
       [0.00333333, 0.00333333, 0.00333333, 0.00333333, 0.00333333,
        0.00333333, 0.00333333, 0.00333333, 0.00333333, 0.00333333],
       [0.00333333, 0.00333333, 0.00333333, 0.00333333, 0.00333333,
        0.00333333, 0.00333333, 0.00333333, 0.00333333, 0.00333333],
       [0.00333333, 0.00333333, 0.00333333, 0.00333333, 0.00333333,
        0.00333333, 0.00333333, 0.00333333, 0.00333333, 0.00333333]])
```

In [59]:

```
np.linalg.matrix_rank(a)
```

Out[59]:

```
1
```

In [60]:

```
np.linalg.lstsq(a,b)
```

```
/Users/dingzheyu/anaconda3/lib/python3.6/site-packages/ipykernel_launc
her.py:1: FutureWarning: `rcond` parameter will change to the default
of machine precision times ``max(M, N)`` where M and N are the input m
atrix dimensions.
To use the future default and silence this warning we advise to pass `
rcond=None`, to keep using the old, explicitly pass `rcond=-1`.
  """Entry point for launching an IPython kernel.
```

Out[60]:

```
(array([[0.14333333, 0.15666667, 0.13333333, 0.19      , 0.20666667,
         0.17666667, 0.15666667, 0.16333333, 0.18666667, 0.17333333],
        [0.14333333, 0.15666667, 0.13333333, 0.19      , 0.20666667,
         0.17666667, 0.15666667, 0.16333333, 0.18666667, 0.17333333],
        [0.14333333, 0.15666667, 0.13333333, 0.19      , 0.20666667,
         0.17666667, 0.15666667, 0.16333333, 0.18666667, 0.17333333],
        [0.14333333, 0.15666667, 0.13333333, 0.19      , 0.20666667,
         0.17666667, 0.15666667, 0.16333333, 0.18666667, 0.17333333],
        [0.14333333, 0.15666667, 0.13333333, 0.19      , 0.20666667,
         0.17666667, 0.15666667, 0.16333333, 0.18666667, 0.17333333],
        [0.14333333, 0.15666667, 0.13333333, 0.19      , 0.20666667,
         0.17666667, 0.15666667, 0.16333333, 0.18666667, 0.17333333],
        [0.14333333, 0.15666667, 0.13333333, 0.19      , 0.20666667,
         0.17666667, 0.15666667, 0.16333333, 0.18666667, 0.17333333],
        [0.14333333, 0.15666667, 0.13333333, 0.19      , 0.20666667,
         0.17666667, 0.15666667, 0.16333333, 0.18666667, 0.17333333],
        [0.14333333, 0.15666667, 0.13333333, 0.19      , 0.20666667,
         0.17666667, 0.15666667, 0.16333333, 0.18666667, 0.17333333],
        [0.14333333, 0.15666667, 0.13333333, 0.19      , 0.20666667,
         0.17666667, 0.15666667, 0.16333333, 0.18666667, 0.1733333
3]]),
 array([], dtype=float64),
 1,
 array([3.00000000e+001, 5.83700565e-016, 8.75811540e-047, 5.48350007e
-078,
        2.62450743e-109, 5.36630706e-142, 2.95853564e-175, 0.00000000e
+000,
        0.00000000e+000, 0.00000000e+000]))
```

In [61]:

```
U, S, Vh = np.linalg.svd(np.random.randint(1, 9, (10, 10, 10)))
V = Vh.T
```

In [62]:

```
np.linalg.cholesky(np.random.randint(1, 2, (1, 1))).T
```

Out[62]:

```
array([[1.]])
```

In [63]:

```
D,V = np.linalg.eig(a)
```

In [64]:

```
D,V = scipy.linalg.eig(a,b)
```

In [65]:

```
Q,R = scipy.linalg.qr(a)
```

In [66]:

```
LU,P = scipy.linalg.lu_factor(a)
```

```
/Users/dingzheyu/anaconda3/lib/python3.6/site-packages/ipykernel_launc
her.py:1: LinAlgWarning: Diagonal number 2 is exactly zero. Singular m
atrix.
  """Entry point for launching an IPython kernel.
```

In [67]:

```
np.fft.fft(a)
```

Out[67]:

```
array([[30.+0.j,  0.+0.j,  0.+0.j,  0.+0.j,  0.+0.j,  0.+0.j,  0.+0.j,
          0.+0.j,  0.+0.j,  0.+0.j],
       [30.+0.j,  0.+0.j,  0.+0.j,  0.+0.j,  0.+0.j,  0.+0.j,  0.+0.j,
          0.+0.j,  0.+0.j,  0.+0.j],
       [30.+0.j,  0.+0.j,  0.+0.j,  0.+0.j,  0.+0.j,  0.+0.j,  0.+0.j,
          0.+0.j,  0.+0.j,  0.+0.j],
       [30.+0.j,  0.+0.j,  0.+0.j,  0.+0.j,  0.+0.j,  0.+0.j,  0.+0.j,
          0.+0.j,  0.+0.j,  0.+0.j],
       [30.+0.j,  0.+0.j,  0.+0.j,  0.+0.j,  0.+0.j,  0.+0.j,  0.+0.j,
          0.+0.j,  0.+0.j,  0.+0.j],
       [30.+0.j,  0.+0.j,  0.+0.j,  0.+0.j,  0.+0.j,  0.+0.j,  0.+0.j,
          0.+0.j,  0.+0.j,  0.+0.j],
       [30.+0.j,  0.+0.j,  0.+0.j,  0.+0.j,  0.+0.j,  0.+0.j,  0.+0.j,
          0.+0.j,  0.+0.j,  0.+0.j],
       [30.+0.j,  0.+0.j,  0.+0.j,  0.+0.j,  0.+0.j,  0.+0.j,  0.+0.j,
          0.+0.j,  0.+0.j,  0.+0.j],
       [30.+0.j,  0.+0.j,  0.+0.j,  0.+0.j,  0.+0.j,  0.+0.j,  0.+0.j,
          0.+0.j,  0.+0.j,  0.+0.j],
       [30.+0.j,  0.+0.j,  0.+0.j,  0.+0.j,  0.+0.j,  0.+0.j,  0.+0.j,
          0.+0.j,  0.+0.j,  0.+0.j]])
```

In [68]:

```
np.fft.ifft(a)
```

Out[68]:

```
array([[3.+0.j, 0.+0.j, 0.+0.j, 0.+0.j, 0.+0.j, 0.+0.j, 0.+0.j, 0.+0.
j,
         0.+0.j, 0.+0.j],
       [3.+0.j, 0.+0.j, 0.+0.j, 0.+0.j, 0.+0.j, 0.+0.j, 0.+0.j, 0.+0.
j,
         0.+0.j, 0.+0.j],
       [3.+0.j, 0.+0.j, 0.+0.j, 0.+0.j, 0.+0.j, 0.+0.j, 0.+0.j, 0.+0.
j,
         0.+0.j, 0.+0.j],
       [3.+0.j, 0.+0.j, 0.+0.j, 0.+0.j, 0.+0.j, 0.+0.j, 0.+0.j, 0.+0.
j,
         0.+0.j, 0.+0.j],
       [3.+0.j, 0.+0.j, 0.+0.j, 0.+0.j, 0.+0.j, 0.+0.j, 0.+0.j, 0.+0.
j,
         0.+0.j, 0.+0.j],
       [3.+0.j, 0.+0.j, 0.+0.j, 0.+0.j, 0.+0.j, 0.+0.j, 0.+0.j, 0.+0.
j,
         0.+0.j, 0.+0.j],
       [3.+0.j, 0.+0.j, 0.+0.j, 0.+0.j, 0.+0.j, 0.+0.j, 0.+0.j, 0.+0.
j,
         0.+0.j, 0.+0.j],
       [3.+0.j, 0.+0.j, 0.+0.j, 0.+0.j, 0.+0.j, 0.+0.j, 0.+0.j, 0.+0.
j,
         0.+0.j, 0.+0.j],
       [3.+0.j, 0.+0.j, 0.+0.j, 0.+0.j, 0.+0.j, 0.+0.j, 0.+0.j, 0.+0.
j,
         0.+0.j, 0.+0.j],
       [3.+0.j, 0.+0.j, 0.+0.j, 0.+0.j, 0.+0.j, 0.+0.j, 0.+0.j, 0.+0.
j,
         0.+0.j, 0.+0.j]])
```

In [69]:

```
np.sort(a)
```

Out[69]:

```
array([[3, 3, 3, 3, 3, 3, 3, 3, 3, 3],
       [3, 3, 3, 3, 3, 3, 3, 3, 3, 3],
       [3, 3, 3, 3, 3, 3, 3, 3, 3, 3],
       [3, 3, 3, 3, 3, 3, 3, 3, 3, 3],
       [3, 3, 3, 3, 3, 3, 3, 3, 3, 3],
       [3, 3, 3, 3, 3, 3, 3, 3, 3, 3],
       [3, 3, 3, 3, 3, 3, 3, 3, 3, 3],
       [3, 3, 3, 3, 3, 3, 3, 3, 3, 3],
       [3, 3, 3, 3, 3, 3, 3, 3, 3, 3],
       [3, 3, 3, 3, 3, 3, 3, 3, 3, 3]])
```

In [70]:

```
I = np.argsort(a[:,1])
```

In [71]:

```
np.linalg.lstsq(a,b)
```

```
/Users/dingzheyu/anaconda3/lib/python3.6/site-packages/ipykernel_launc
her.py:1: FutureWarning: `rcond` parameter will change to the default
of machine precision times ``max(M, N)`` where M and N are the input m
atrix dimensions.
To use the future default and silence this warning we advise to pass `
rcond=None`, to keep using the old, explicitly pass `rcond=-1`.
  """Entry point for launching an IPython kernel.
```

Out[71]:

```
(array([[0.14333333, 0.15666667, 0.13333333, 0.19      , 0.20666667,
         0.17666667, 0.15666667, 0.16333333, 0.18666667, 0.17333333],
        [0.14333333, 0.15666667, 0.13333333, 0.19      , 0.20666667,
         0.17666667, 0.15666667, 0.16333333, 0.18666667, 0.17333333],
        [0.14333333, 0.15666667, 0.13333333, 0.19      , 0.20666667,
         0.17666667, 0.15666667, 0.16333333, 0.18666667, 0.17333333],
        [0.14333333, 0.15666667, 0.13333333, 0.19      , 0.20666667,
         0.17666667, 0.15666667, 0.16333333, 0.18666667, 0.17333333],
        [0.14333333, 0.15666667, 0.13333333, 0.19      , 0.20666667,
         0.17666667, 0.15666667, 0.16333333, 0.18666667, 0.17333333],
        [0.14333333, 0.15666667, 0.13333333, 0.19      , 0.20666667,
         0.17666667, 0.15666667, 0.16333333, 0.18666667, 0.17333333],
        [0.14333333, 0.15666667, 0.13333333, 0.19      , 0.20666667,
         0.17666667, 0.15666667, 0.16333333, 0.18666667, 0.17333333],
        [0.14333333, 0.15666667, 0.13333333, 0.19      , 0.20666667,
         0.17666667, 0.15666667, 0.16333333, 0.18666667, 0.17333333],
        [0.14333333, 0.15666667, 0.13333333, 0.19      , 0.20666667,
         0.17666667, 0.15666667, 0.16333333, 0.18666667, 0.17333333],
        [0.14333333, 0.15666667, 0.13333333, 0.19      , 0.20666667,
         0.17666667, 0.15666667, 0.16333333, 0.18666667, 0.1733333
3]]),
 array([], dtype=float64),
 1,
 array([3.00000000e+001, 5.83700565e-016, 8.75811540e-047, 5.48350007e
-078,
        2.62450743e-109, 5.36630706e-142, 2.95853564e-175, 0.00000000e
+000,
        0.00000000e+000, 0.00000000e+000]))
```

In [72]:

```
np.unique(a)
```

Out[72]:

```
array([3])
```

In [73]:

```
a.squeeze()
```

Out[73]:

```
array([[3, 3, 3, 3, 3, 3, 3, 3, 3, 3],
       [3, 3, 3, 3, 3, 3, 3, 3, 3, 3],
       [3, 3, 3, 3, 3, 3, 3, 3, 3, 3],
       [3, 3, 3, 3, 3, 3, 3, 3, 3, 3],
       [3, 3, 3, 3, 3, 3, 3, 3, 3, 3],
       [3, 3, 3, 3, 3, 3, 3, 3, 3, 3],
       [3, 3, 3, 3, 3, 3, 3, 3, 3, 3],
       [3, 3, 3, 3, 3, 3, 3, 3, 3, 3],
       [3, 3, 3, 3, 3, 3, 3, 3, 3, 3],
       [3, 3, 3, 3, 3, 3, 3, 3, 3, 3]])
```

Task 3

In [74]:

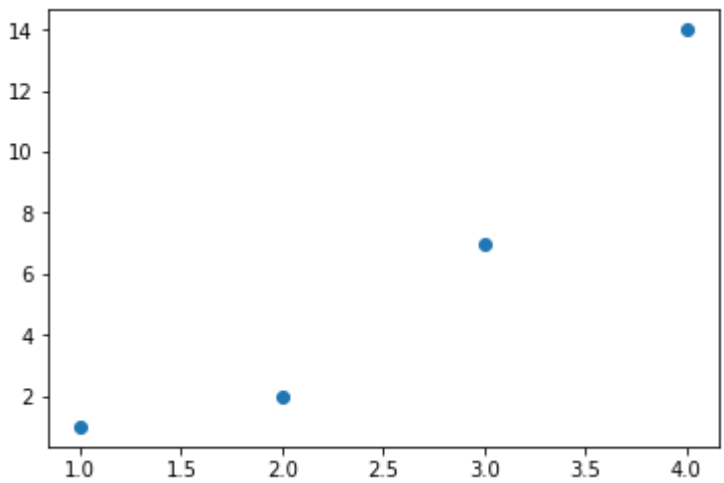
```
import matplotlib.pyplot as plt
plt.plot([1,2,3,4], [1,2,7,14])
plt.axis([0, 6, 0, 20])
plt.show()
```

```
<Figure size 640x480 with 1 Axes>
```

Task 4

In [75]:

```
plt.scatter([1,2,3,4], [1,2,7,14])
plt.show()
```



Task 5

Acccount name: BenDing96

Task 6

https://github.com/BenDing96/Deep-Learning (https://github.com/BenDing96/Deep-Learning)

In [ ]: