Math 4430 Project Python Code Zijiang Yan

December 4, 2020

[1]:
```
import numpy as np
import random
from __future__ import division


Q_1 = np.zeros((20,20))


for i in range(0, 20, 1):
    for j in range(0, 20, 1):

            Q_1[i][j]=(1/20)

#print(Q_1)
print("There are 5 pairs forbidden cities")
for k in range(0,5,1):
    value1 = random.randint(0, 20)  #random pick 1 city in the city_list.
    value2 = random.randint(0, 20)  #random pick 1 city in the city_list.
    # we need to make sure different city
    while(value1 == value2):
        value2 = random.randint(0, 20)
    print("The forbiden pair of cities index ", k, " is ",value1,value2)
    for i in range(0, 20, 1):
        for j in range(0, 20, 1):

            if(i == j):
                Q_1[i][j] = 0 # It is not possible if we departure and arrive in␣
 ↪the same city for single route


            #  otherwise, forbidden city pairs should be mark as 0.
        Q_1[value1-1][value2-1] = 0 # value1 to value2 is forbidden
        Q_1[value2-1][value1-1] = 0 #the counterpart is forbidden
#print(Q_1)

def count_non_zero(row):
```

```
    # if transition state is not 0 , means it has the probability to approach␣
 ↪the other city
    # for example {0.5,0.5,0,0,0,....} the result should be 2
    count= 0
    for j in range(0, 20, 1):
        if  Q_1[row][j] != 0:
            count = count +1

    return count # count is non-zero result

for i in range(0, 20, 1):
    non_zero_count = count_non_zero(i)
    for j in range(0, 20, 1):
        if  Q_1[i][j] != 0: # if there is not in the same city
                Q_1[i][j] = 1/non_zero_count # we redefine the Q matrix make␣
 ↪non-zero probability are uniform.


print(Q_1)  # old transition matrix
Q =Q_1
```

```
There are 5 pairs forbidden cities
The forbiden pair of cities index  0  is  6 14
The forbiden pair of cities index  1  is  16 13
The forbiden pair of cities index  2  is  5 13
The forbiden pair of cities index  3  is  4 10
The forbiden pair of cities index  4  is  4 9
[[0.         0.05263158 0.05263158 0.05263158 0.05263158 0.05263158
  0.05263158 0.05263158 0.05263158 0.05263158 0.05263158 0.05263158
  0.05263158 0.05263158 0.05263158 0.05263158 0.05263158 0.05263158
  0.05263158 0.05263158]
 [0.05263158 0.         0.05263158 0.05263158 0.05263158 0.05263158
  0.05263158 0.05263158 0.05263158 0.05263158 0.05263158 0.05263158
  0.05263158 0.05263158 0.05263158 0.05263158 0.05263158 0.05263158
  0.05263158 0.05263158]
 [0.05263158 0.05263158 0.         0.05263158 0.05263158 0.05263158
  0.05263158 0.05263158 0.05263158 0.05263158 0.05263158 0.05263158
  0.05263158 0.05263158 0.05263158 0.05263158 0.05263158 0.05263158
  0.05263158 0.05263158]
 [0.05882353 0.05882353 0.05882353 0.         0.05882353 0.05882353
  0.05882353 0.05882353 0.         0.         0.05882353 0.05882353
  0.05882353 0.05882353 0.05882353 0.05882353 0.05882353 0.05882353
  0.05882353 0.05882353]
 [0.05555556 0.05555556 0.05555556 0.05555556 0.         0.05555556
  0.05555556 0.05555556 0.05555556 0.05555556 0.05555556 0.05555556
  0.         0.05555556 0.05555556 0.05555556 0.05555556 0.05555556
  0.05555556 0.05555556]
```

```
[0.05555556 0.05555556 0.05555556 0.05555556 0.05555556 0.
 0.05555556 0.05555556 0.05555556 0.05555556 0.05555556 0.05555556
 0.05555556 0.         0.05555556 0.05555556 0.05555556 0.05555556
 0.05555556 0.05555556]
[0.05263158 0.05263158 0.05263158 0.05263158 0.05263158 0.05263158
 0.         0.05263158 0.05263158 0.05263158 0.05263158 0.05263158
 0.05263158 0.05263158 0.05263158 0.05263158 0.05263158 0.05263158
 0.05263158 0.05263158]
[0.05263158 0.05263158 0.05263158 0.05263158 0.05263158 0.05263158
 0.05263158 0.         0.05263158 0.05263158 0.05263158 0.05263158
 0.05263158 0.05263158 0.05263158 0.05263158 0.05263158 0.05263158
 0.05263158 0.05263158]
[0.05555556 0.05555556 0.05555556 0.         0.05555556 0.05555556
 0.05555556 0.05555556 0.         0.05555556 0.05555556 0.05555556
 0.05555556 0.05555556 0.05555556 0.05555556 0.05555556 0.05555556
 0.05555556 0.05555556]
[0.05555556 0.05555556 0.05555556 0.         0.05555556 0.05555556
 0.05555556 0.05555556 0.05555556 0.         0.05555556 0.05555556
 0.05555556 0.05555556 0.05555556 0.05555556 0.05555556 0.05555556
 0.05555556 0.05555556]
[0.05263158 0.05263158 0.05263158 0.05263158 0.05263158 0.05263158
 0.05263158 0.05263158 0.05263158 0.05263158 0.         0.05263158
 0.05263158 0.05263158 0.05263158 0.05263158 0.05263158 0.05263158
 0.05263158 0.05263158]
[0.05263158 0.05263158 0.05263158 0.05263158 0.05263158 0.05263158
 0.05263158 0.05263158 0.05263158 0.05263158 0.05263158 0.
 0.05263158 0.05263158 0.05263158 0.05263158 0.05263158 0.05263158
 0.05263158 0.05263158]
[0.05882353 0.05882353 0.05882353 0.05882353 0.         0.05882353
 0.05882353 0.05882353 0.05882353 0.05882353 0.05882353 0.05882353
 0.         0.05882353 0.05882353 0.         0.05882353 0.05882353
 0.05882353 0.05882353]
[0.05555556 0.05555556 0.05555556 0.05555556 0.05555556 0.
 0.05555556 0.05555556 0.05555556 0.05555556 0.05555556 0.05555556
 0.05555556 0.         0.05555556 0.05555556 0.05555556 0.05555556
 0.05555556 0.05555556]
[0.05263158 0.05263158 0.05263158 0.05263158 0.05263158 0.05263158
 0.05263158 0.05263158 0.05263158 0.05263158 0.05263158 0.05263158
 0.05263158 0.05263158 0.         0.05263158 0.05263158 0.05263158
 0.05263158 0.05263158]
[0.05555556 0.05555556 0.05555556 0.05555556 0.05555556 0.05555556
 0.05555556 0.05555556 0.05555556 0.05555556 0.05555556 0.05555556
 0.         0.05555556 0.05555556 0.         0.05555556 0.05555556
 0.05555556 0.05555556]
[0.05263158 0.05263158 0.05263158 0.05263158 0.05263158 0.05263158
 0.05263158 0.05263158 0.05263158 0.05263158 0.05263158 0.05263158
 0.05263158 0.05263158 0.05263158 0.05263158 0.         0.05263158
 0.05263158 0.05263158]
```

```
[0.05263158 0.05263158 0.05263158 0.05263158 0.05263158 0.05263158
 0.05263158 0.05263158 0.05263158 0.05263158 0.05263158 0.05263158
 0.05263158 0.05263158 0.05263158 0.05263158 0.05263158 0.
 0.05263158 0.05263158]
[0.05263158 0.05263158 0.05263158 0.05263158 0.05263158 0.05263158
 0.05263158 0.05263158 0.05263158 0.05263158 0.05263158 0.05263158
 0.05263158 0.05263158 0.05263158 0.05263158 0.05263158 0.05263158
 0.         0.05263158]
[0.05263158 0.05263158 0.05263158 0.05263158 0.05263158 0.05263158
 0.05263158 0.05263158 0.05263158 0.05263158 0.05263158 0.05263158
 0.05263158 0.05263158 0.05263158 0.05263158 0.05263158 0.05263158
 0.05263158 0.        ]]
```

[2]:
```
Q_2 = np.zeros((20,20)) #travel distance matrix
for i in range(0, 20, 1):
    for j in range(0, 20, 1):
        # 1/20 no forbidde
            if(i == j):
                Q_2[i][j] = 0 # depature and arrival are same city ,mark is as 0

            else:
                value = random.randint(1, 100)  #find random distance from 0 to␣
 ↪100.
                Q_2[i][j] = value # assign the distance to the i,j entry.

#print(Q_2)

for i in range(0, 20, 1):
    for j in range(0, 20, 1):
        if Q_2[i][j] != 0:
            Q_2[j][i] = Q_2[i][j] #  we need to make sure the distance from city␣
 ↪A to city B is the same as the city B to city A
        #make sure it is symmetry
print(Q_2)
```

```
[[  0.  44.  10.  63.  31.  65.  30.  59.   4.  90.  94.  99.  78.  76.
   21.  45.  23.  27.  19.  13.]
 [ 44.   0.  19.  83.  37.  90.  59.  90.  76.  86.  29.  64.  58.  53.
   92.  58.  33.  40.  84.  71.]
 [ 10.  19.   0.  92.  47.  33.  89.  18.  31.  98.  24.  78.  20.   5.
   17.  54.  80.  60.  32.  25.]
 [ 63.  83.  92.   0.  35.  12.  20.  29.  57.  21.  47.  55.  93.  55.
   30.  23.  30.  54.  96.  34.]
 [ 31.  37.  47.  35.   0.  99.  76.  53.  72.  96.  27.  12.  55.  38.
   27.  78.  40.  38.  35.  30.]
 [ 65.  90.  33.  12.  99.   0.  48.  83.   5.  17.  50.  52.  81.  11.
    7.  71.  49.  80.   8.  60.]
 [ 30.  59.  89.  20.  76.  48.   0.  18.  85.  40.  45.  59.  78.  74.
```

```
  63.  46.  91.  96.  46.  87.]
[ 59.  90.  18.  29.  53.  83.  18.   0.  54.  21.  33.  31.  92.  39.
  73.  25.  11.  98.  20.  35.]
[  4.  76.  31.  57.  72.   5.  85.  54.   0.  22.   7.  41.  40. 100.
  70.  33.  42.  77.  82.  86.]
[ 90.  86.  98.  21.  96.  17.  40.  21.  22.   0.  98.  98.  34.  80.
  85.  18.  81.  30.   8.  55.]
[ 94.  29.  24.  47.  27.  50.  45.  33.   7.  98.   0.  80.  83.  72.
  88.  22.  84.  85.  71.  57.]
[ 99.  64.  78.  55.  12.  52.  59.  31.  41.  98.  80.   0.  75.  70.
  94.  85.  33.  96.   6.  82.]
[ 78.  58.  20.  93.  55.  81.  78.  92.  40.  34.  83.  75.   0.  37.
  92.  22.  73.  84.  86.  33.]
[ 76.  53.   5.  55.  38.  11.  74.  39. 100.  80.  72.  70.  37.   0.
  18.  83.  59.  56.  94.  65.]
[ 21.  92.  17.  30.  27.   7.  63.  73.  70.  85.  88.  94.  92.  18.
   0.   6.  95.  16.  31.  35.]
[ 45.  58.  54.  23.  78.  71.  46.  25.  33.  18.  22.  85.  22.  83.
   6.   0.  89.  65.   4.  50.]
[ 23.  33.  80.  30.  40.  49.  91.  11.  42.  81.  84.  33.  73.  59.
  95.  89.   0.   5.   6.  29.]
[ 27.  40.  60.  54.  38.  80.  96.  98.  77.  30.  85.  96.  84.  56.
  16.  65.   5.   0.  88.  41.]
[ 19.  84.  32.  96.  35.   8.  46.  20.  82.   8.  71.   6.  86.  94.
  31.   4.   6.  88.   0.  99.]
[ 13.  71.  25.  34.  30.  60.  87.  35.  86.  55.  57.  82.  33.  65.
  35.  50.  29.  41.  99.   0.]]
```

[3]:
```
# Metropolis Algorithm
pi=[] # we construct a uniform initial distributution
for j in range(0, 20, 1):
        # 1/20 no forbidde
        pi.append(1/20)
print("Initial Distribution ",pi)
```

```
Initial Distribution  [0.05, 0.05, 0.05, 0.05, 0.05, 0.05, 0.05, 0.05, 0.05,
0.05, 0.05, 0.05, 0.05, 0.05, 0.05, 0.05, 0.05, 0.05, 0.05, 0.05]
```

[4]:
```
from __future__ import division
import math

# this part is simply compute  alpha = min(1, miu_k/miu_j)
def alpha (r,c):#r,c from 0-4
    r = r-1
    c = c-1
    if((pi[r]*Q[r][c])==0):
        x = -1 #Denominator is 0 error
    else:
```

```
        x = (pi[c]*Q[c][r]* 1.0)/(pi[r]*Q[r][c])

    if(x>1):
        result = 1
    elif (x > 0): # 0<x<1
        result = x
    else:
        result = -1 #error
    return result
```

[5]:
```
def new_transition(i,j):#compute  new transition matrix
    i = i-1
    j = j-1
    result  =-1 #error
    #if r != c:
    if alpha(i,j) != -1: #if i != j:
        result = Q[i][j] * alpha(i+1,j+1) *1.0
    else: # if i == j
        result  = 1
        for k in range(0, len(Q), 1):
            if k != i:
                result = result - Q[i][k] * alpha(i+1,k+1) *1.0

    return result


for i in range(1, len(pi)+1, 1): #compute diagonal of new transition matrix P
    print("fnewTransition(%d,%d)is%f",i,i,new_transition(i,i))
```

```
fnewTransition(%d,%d)is%f 1 1 5.551115123125783e-16
fnewTransition(%d,%d)is%f 2 2 5.551115123125783e-16
fnewTransition(%d,%d)is%f 3 3 5.551115123125783e-16
fnewTransition(%d,%d)is%f 4 4 0.08737530099759233
fnewTransition(%d,%d)is%f 5 5 0.035087719298245806
fnewTransition(%d,%d)is%f 6 6 0.035087719298245806
fnewTransition(%d,%d)is%f 7 7 5.551115123125783e-16
fnewTransition(%d,%d)is%f 8 8 5.551115123125783e-16
fnewTransition(%d,%d)is%f 9 9 0.03508771929824592
fnewTransition(%d,%d)is%f 10 10 0.03508771929824592
fnewTransition(%d,%d)is%f 11 11 5.551115123125783e-16
fnewTransition(%d,%d)is%f 12 12 5.551115123125783e-16
fnewTransition(%d,%d)is%f 13 13 0.08737530099759222
fnewTransition(%d,%d)is%f 14 14 0.035087719298245806
fnewTransition(%d,%d)is%f 15 15 5.551115123125783e-16
fnewTransition(%d,%d)is%f 16 16 0.03508771929824578
fnewTransition(%d,%d)is%f 17 17 5.551115123125783e-16
fnewTransition(%d,%d)is%f 18 18 5.551115123125783e-16
fnewTransition(%d,%d)is%f 19 19 5.551115123125783e-16
```

```
fnewTransition(%d,%d)is%f 20 20 5.551115123125783e-16
```

[10]:
```
A = np.zeros((len(Q),len(Q[0])))

for i in range(0, len(Q), 1):
    for j in range(0, len(Q[0]), 1):
            A[i][j] = new_transition(i,j)
display("Initial transition matrix")
print(A)
from numpy.linalg import matrix_power
display("transition matrix Repeating 10 times")
A_2 =matrix_power(A, 10) #This is the new Transition Matrix after 10 repetitions.
print(A_2)

display("transition matrix Repeating 20 times")
A_2 =matrix_power(A, 20) #This is the new Transition Matrix after 10 repetitions.
print(A_2)
#A_100 =matrix_power(A, 100)
#print(A_100)
```

```
'Initial transition matrix'


[[ 5.55111512e-16  5.26315789e-02  5.26315789e-02  5.26315789e-02
   5.26315789e-02  5.26315789e-02  5.26315789e-02  5.26315789e-02
   5.26315789e-02  5.26315789e-02  5.26315789e-02  5.26315789e-02
   5.26315789e-02  5.26315789e-02  5.26315789e-02  5.26315789e-02
   5.26315789e-02  5.26315789e-02  5.26315789e-02  5.26315789e-02]
 [ 5.26315789e-02  5.55111512e-16  5.26315789e-02  5.26315789e-02
   5.26315789e-02  5.26315789e-02  5.26315789e-02  5.26315789e-02
   5.26315789e-02  5.26315789e-02  5.26315789e-02  5.26315789e-02
   5.26315789e-02  5.26315789e-02  5.26315789e-02  5.26315789e-02
   5.26315789e-02  5.26315789e-02  5.26315789e-02  5.26315789e-02]
 [ 5.26315789e-02  5.26315789e-02  5.55111512e-16  5.26315789e-02
   5.26315789e-02  5.26315789e-02  5.26315789e-02  5.26315789e-02
   5.26315789e-02  5.26315789e-02  5.26315789e-02  5.26315789e-02
   5.26315789e-02  5.26315789e-02  5.26315789e-02  5.26315789e-02
   5.26315789e-02  5.26315789e-02  5.26315789e-02  5.26315789e-02]
 [ 5.26315789e-02  5.26315789e-02  5.26315789e-02  5.55111512e-16
   5.26315789e-02  5.26315789e-02  5.26315789e-02  5.26315789e-02
   5.26315789e-02  5.26315789e-02  5.26315789e-02  5.26315789e-02
   5.26315789e-02  5.26315789e-02  5.26315789e-02  5.26315789e-02
   5.26315789e-02  5.26315789e-02  5.26315789e-02  5.26315789e-02]
 [ 5.26315789e-02  5.26315789e-02  5.26315789e-02  5.26315789e-02
   8.73753010e-02  5.55555556e-02  5.55555556e-02  5.26315789e-02
   5.26315789e-02 -0.00000000e+00 -0.00000000e+00  5.26315789e-02
   5.26315789e-02  5.88235294e-02  5.55555556e-02  5.26315789e-02
   5.55555556e-02  5.26315789e-02  5.26315789e-02  5.26315789e-02]
 [ 5.26315789e-02  5.26315789e-02  5.26315789e-02  5.26315789e-02
```

```
  5.55555556e-02  3.50877193e-02  5.55555556e-02  5.26315789e-02
  5.26315789e-02  5.55555556e-02  3.50877193e-02  3.50877193e-02
  5.26315789e-02 -0.00000000e+00  5.55555556e-02  5.26315789e-02
  5.55555556e-02  5.26315789e-02  5.26315789e-02  5.26315789e-02]
[ 5.26315789e-02  5.26315789e-02  5.26315789e-02  5.26315789e-02
  5.55555556e-02  5.55555556e-02  3.50877193e-02  5.26315789e-02
  5.26315789e-02  5.55555556e-02  5.55555556e-02  5.26315789e-02
  5.26315789e-02  5.55555556e-02  3.50877193e-02  5.26315789e-02
  5.55555556e-02  5.26315789e-02  5.26315789e-02  5.26315789e-02]
[ 5.26315789e-02  5.26315789e-02  5.26315789e-02  5.26315789e-02
  5.26315789e-02  5.26315789e-02  5.26315789e-02  5.55111512e-16
  5.26315789e-02  5.26315789e-02  5.26315789e-02  5.26315789e-02
  5.26315789e-02  5.26315789e-02  5.26315789e-02  5.55111512e-16
  5.26315789e-02  5.26315789e-02  5.26315789e-02  5.26315789e-02]
[ 5.26315789e-02  5.26315789e-02  5.26315789e-02  5.26315789e-02
  5.26315789e-02  5.26315789e-02  5.26315789e-02  5.26315789e-02
  5.55111512e-16  5.26315789e-02  5.26315789e-02  5.26315789e-02
  5.26315789e-02  5.26315789e-02  5.26315789e-02  5.26315789e-02
  5.26315789e-02  5.26315789e-02  5.26315789e-02  5.26315789e-02]
[ 5.26315789e-02  5.26315789e-02  5.26315789e-02  5.26315789e-02
 -0.00000000e+00  5.55555556e-02  5.55555556e-02  5.26315789e-02
  5.26315789e-02  3.50877193e-02  5.55555556e-02  5.26315789e-02
  5.26315789e-02  5.55555556e-02  5.55555556e-02  5.26315789e-02
  5.55555556e-02  5.26315789e-02  5.26315789e-02  5.26315789e-02]
[ 5.26315789e-02  5.26315789e-02  5.26315789e-02  5.26315789e-02
 -0.00000000e+00  3.50877193e-02  5.55555556e-02  5.26315789e-02
  5.26315789e-02  5.55555556e-02  3.50877193e-02  5.26315789e-02
  5.26315789e-02  5.55555556e-02  5.55555556e-02  5.26315789e-02
  5.55555556e-02  5.26315789e-02  5.26315789e-02  5.26315789e-02]
[ 5.26315789e-02  5.26315789e-02  5.26315789e-02  5.26315789e-02
  5.26315789e-02  5.55111512e-16  5.26315789e-02  5.26315789e-02
  5.26315789e-02  5.26315789e-02  5.26315789e-02  5.55111512e-16
  5.26315789e-02  5.26315789e-02  5.26315789e-02  5.26315789e-02
  5.26315789e-02  5.26315789e-02  5.26315789e-02  5.26315789e-02]
[ 5.26315789e-02  5.26315789e-02  5.26315789e-02  5.26315789e-02
  5.26315789e-02  5.26315789e-02  5.26315789e-02  5.26315789e-02
  5.26315789e-02  5.26315789e-02  5.26315789e-02  5.26315789e-02
  5.55111512e-16  5.26315789e-02  5.26315789e-02  5.26315789e-02
  5.26315789e-02  5.26315789e-02  5.26315789e-02  5.26315789e-02]
[ 5.26315789e-02  5.26315789e-02  5.26315789e-02  5.26315789e-02
  5.88235294e-02 -0.00000000e+00  5.55555556e-02  5.26315789e-02
  5.26315789e-02  5.55555556e-02  5.55555556e-02  5.26315789e-02
  5.26315789e-02  8.73753010e-02  5.55555556e-02  5.26315789e-02
 -0.00000000e+00  5.26315789e-02  5.26315789e-02  5.26315789e-02]
[ 5.26315789e-02  5.26315789e-02  5.26315789e-02  5.26315789e-02
  5.55555556e-02  5.55555556e-02  3.50877193e-02  5.26315789e-02
  5.26315789e-02  5.55555556e-02  5.55555556e-02  5.26315789e-02
  5.26315789e-02  5.55555556e-02  3.50877193e-02  5.26315789e-02
```

```
  5.55555556e-02  3.50877193e-02  5.26315789e-02  5.26315789e-02]
 [ 5.26315789e-02  5.26315789e-02  5.26315789e-02  5.26315789e-02
   5.26315789e-02  5.26315789e-02  5.26315789e-02  5.55111512e-16
   5.26315789e-02  5.26315789e-02  5.26315789e-02  5.26315789e-02
   5.26315789e-02  5.26315789e-02  5.26315789e-02  5.55111512e-16
   5.26315789e-02  5.26315789e-02  5.26315789e-02  5.26315789e-02]
 [ 5.26315789e-02  5.26315789e-02  5.26315789e-02  5.26315789e-02
   5.55555556e-02  5.55555556e-02  5.55555556e-02  5.26315789e-02
   5.26315789e-02  5.55555556e-02  5.55555556e-02  5.26315789e-02
   5.26315789e-02 -0.00000000e+00  5.55555556e-02  5.26315789e-02
   3.50877193e-02  5.26315789e-02  5.26315789e-02  5.26315789e-02]
 [ 5.26315789e-02  5.26315789e-02  5.26315789e-02  5.26315789e-02
   5.26315789e-02  5.26315789e-02  5.26315789e-02  5.26315789e-02
   5.26315789e-02  5.26315789e-02  5.26315789e-02  5.26315789e-02
   5.26315789e-02  5.26315789e-02  5.55111512e-16  5.26315789e-02
   5.26315789e-02  5.55111512e-16  5.26315789e-02  5.26315789e-02]
 [ 5.26315789e-02  5.26315789e-02  5.26315789e-02  5.26315789e-02
   5.26315789e-02  5.26315789e-02  5.26315789e-02  5.26315789e-02
   5.26315789e-02  5.26315789e-02  5.26315789e-02  5.26315789e-02
   5.26315789e-02  5.26315789e-02  5.26315789e-02  5.26315789e-02
   5.26315789e-02  5.26315789e-02  5.55111512e-16  5.26315789e-02]
 [ 5.26315789e-02  5.26315789e-02  5.26315789e-02  5.26315789e-02
   5.26315789e-02  5.26315789e-02  5.26315789e-02  5.26315789e-02
   5.26315789e-02  5.26315789e-02  5.26315789e-02  5.26315789e-02
   5.26315789e-02  5.26315789e-02  5.26315789e-02  5.26315789e-02
   5.26315789e-02  5.26315789e-02  5.26315789e-02  5.55111512e-16]]

'transition matrix Repeating 10 times'


[[0.04592766 0.04592766 0.04592766 0.04592766 0.04597006 0.04263984
  0.0475301  0.04371989 0.04592766 0.04591616 0.04505213 0.04520991
  0.04592766 0.04610995 0.0451279  0.04371989 0.04590846 0.04516803
  0.04592766 0.04592766]
 [0.04592766 0.04592766 0.04592766 0.04592766 0.04597006 0.04263984
  0.0475301  0.04371989 0.04592766 0.04591616 0.04505213 0.04520991
  0.04592766 0.04610995 0.0451279  0.04371989 0.04590846 0.04516803
  0.04592766 0.04592766]
 [0.04592766 0.04592766 0.04592766 0.04592766 0.04597006 0.04263984
  0.0475301  0.04371989 0.04592766 0.04591616 0.04505213 0.04520991
  0.04592766 0.04610995 0.0451279  0.04371989 0.04590846 0.04516803
  0.04592766 0.04592766]
 [0.04592766 0.04592766 0.04592766 0.04592766 0.04597006 0.04263984
  0.0475301  0.04371989 0.04592766 0.04591616 0.04505213 0.04520991
  0.04592766 0.04610995 0.0451279  0.04371989 0.04590846 0.04516803
  0.04592766 0.04592766]
 [0.0459802  0.0459802  0.0459802  0.0459802  0.04602266 0.04268863
  0.04758448 0.04376991 0.0459802  0.04596869 0.04510367 0.04526164
  0.0459802  0.04616271 0.04517953 0.04376991 0.04596099 0.04521971
```

```
 0.0459802  0.0459802 ]
[0.04425083 0.04425083 0.04425083 0.04425083 0.04429169 0.04108306
 0.04579477 0.04212367 0.04425083 0.04423975 0.04340727 0.04355929
 0.04425083 0.04442647 0.04348027 0.04212367 0.04423234 0.04351894
 0.04425083 0.04425083]
[0.04750658 0.04750658 0.04750658 0.04750658 0.04755044 0.04410573
 0.04916411 0.04522291 0.04750658 0.04749468 0.04660095 0.04676416
 0.04750658 0.04769514 0.04667932 0.04522291 0.04748672 0.04672084
 0.04750658 0.04750658]
[0.04371989 0.04371989 0.04371989 0.04371989 0.04376026 0.04059012
 0.0452453  0.04161825 0.04371989 0.04370894 0.04288645 0.04303665
 0.04371989 0.04389342 0.04295857 0.04161825 0.04370162 0.04299678
 0.04371989 0.04371989]
[0.04592766 0.04592766 0.04592766 0.04592766 0.04597006 0.04263984
 0.0475301  0.04371989 0.04592766 0.04591616 0.04505213 0.04520991
 0.04592766 0.04610995 0.0451279  0.04371989 0.04590846 0.04516803
 0.04592766 0.04592766]
[0.04592462 0.04592462 0.04592462 0.04592462 0.04596703 0.04263703
 0.04752696 0.043717   0.04592462 0.04591313 0.04504915 0.04520693
 0.04592462 0.04610691 0.04512492 0.043717   0.04590543 0.04516505
 0.04592462 0.04592462]
[0.04502795 0.04502795 0.04502795 0.04502795 0.04506952 0.04180454
 0.046599   0.04286343 0.04502795 0.04501668 0.04416957 0.04432427
 0.04502795 0.04520667 0.04424386 0.04286343 0.04500913 0.04428321
 0.04502795 0.04502795]
[0.04369308 0.04369308 0.04369308 0.04369308 0.04373342 0.04056523
 0.04521756 0.04159272 0.04369308 0.04368214 0.04286014 0.04301025
 0.04369308 0.0438665  0.04293223 0.04159272 0.04367482 0.04297041
 0.04369308 0.04369308]
[0.04592766 0.04592766 0.04592766 0.04592766 0.04597006 0.04263984
 0.0475301  0.04371989 0.04592766 0.04591616 0.04505213 0.04520991
 0.04592766 0.04610995 0.0451279  0.04371989 0.04590846 0.04516803
 0.04592766 0.04592766]
[0.04602527 0.04602527 0.04602527 0.04602527 0.04606776 0.04273046
 0.04763112 0.04381281 0.04602527 0.04601374 0.04514788 0.045306
 0.04602527 0.04620795 0.04522381 0.04381281 0.04600603 0.04526403
 0.04602527 0.04602527]
[0.04673421 0.04673421 0.04673421 0.04673421 0.04677736 0.04338866
 0.0483648  0.04448767 0.04673421 0.04672251 0.04584331 0.04600387
 0.04673421 0.04691971 0.04592041 0.04448767 0.04671468 0.04596125
 0.04673421 0.04673421]
[0.04371989 0.04371989 0.04371989 0.04371989 0.04376026 0.04059012
 0.0452453  0.04161825 0.04371989 0.04370894 0.04288645 0.04303665
 0.04371989 0.04389342 0.04295857 0.04161825 0.04370162 0.04299678
 0.04371989 0.04371989]
[0.04592215 0.04592215 0.04592215 0.04592215 0.04596455 0.04263473
 0.0475244  0.04371464 0.04592215 0.04591065 0.04504672 0.04520449
 0.04592215 0.04610442 0.04512248 0.04371464 0.04590295 0.04516261
```

```
  0.04592215 0.04592215]
 [0.04356767 0.04356767 0.04356767 0.04356767 0.0436079  0.0404488
  0.04508777 0.04147335 0.04356767 0.04355676 0.04273713 0.04288681
  0.04356767 0.0437406  0.04280901 0.04147335 0.04354946 0.04284708
  0.04356767 0.04356767]
 [0.04592766 0.04592766 0.04592766 0.04592766 0.04597006 0.04263984
  0.0475301  0.04371989 0.04592766 0.04591616 0.04505213 0.04520991
  0.04592766 0.04610995 0.0451279  0.04371989 0.04590846 0.04516803
  0.04592766 0.04592766]
 [0.04592766 0.04592766 0.04592766 0.04592766 0.04597006 0.04263984
  0.0475301  0.04371989 0.04592766 0.04591616 0.04505213 0.04520991
  0.04592766 0.04610995 0.0451279  0.04371989 0.04590846 0.04516803
  0.04592766 0.04592766]]
```

```
'transition matrix Repeating 20 times'
```

```
[[0.04137618 0.04137618 0.04137618 0.04137618 0.04141438 0.03841419
  0.04281982 0.0393872  0.04137618 0.04136582 0.04058741 0.04072956
  0.04137618 0.04154041 0.04065567 0.0393872  0.04135889 0.04069183
  0.04137618 0.04137618]
 [0.04137618 0.04137618 0.04137618 0.04137618 0.04141438 0.03841419
  0.04281982 0.0393872  0.04137618 0.04136582 0.04058741 0.04072956
  0.04137618 0.04154041 0.04065567 0.0393872  0.04135889 0.04069183
  0.04137618 0.04137618]
 [0.04137618 0.04137618 0.04137618 0.04137618 0.04141438 0.03841419
  0.04281982 0.0393872  0.04137618 0.04136582 0.04058741 0.04072956
  0.04137618 0.04154041 0.04065567 0.0393872  0.04135889 0.04069183
  0.04137618 0.04137618]
 [0.04137618 0.04137618 0.04137618 0.04137618 0.04141438 0.03841419
  0.04281982 0.0393872  0.04137618 0.04136582 0.04058741 0.04072956
  0.04137618 0.04154041 0.04065567 0.0393872  0.04135889 0.04069183
  0.04137618 0.04137618]
 [0.04142351 0.04142351 0.04142351 0.04142351 0.04146176 0.03845814
  0.04286881 0.03943226 0.04142351 0.04141314 0.04063385 0.04077616
  0.04142351 0.04158793 0.04070219 0.03943226 0.0414062  0.04073839
  0.04142351 0.04142351]
 [0.03986553 0.03986553 0.03986553 0.03986553 0.03990234 0.03701168
  0.04125646 0.03794917 0.03986553 0.03985555 0.03910556 0.03924252
  0.03986553 0.04002376 0.03917133 0.03794917 0.03984887 0.03920617
  0.03986553 0.03986553]
 [0.04279862 0.04279862 0.04279862 0.04279862 0.04283814 0.03973481
  0.04429189 0.04074127 0.04279862 0.04278791 0.04198274 0.04212978
  0.04279862 0.0429685  0.04205335 0.04074127 0.04278074 0.04209075
  0.04279862 0.04279862]
 [0.0393872  0.0393872  0.0393872  0.0393872  0.03942357 0.0365676
  0.04076144 0.03749383 0.0393872  0.03937734 0.03863635 0.03877167
  0.0393872  0.03954354 0.03870133 0.03749383 0.03937074 0.03873575
  0.0393872  0.0393872 ]
```

```
[0.04137618 0.04137618 0.04137618 0.04137618 0.04141438 0.03841419
 0.04281982 0.0393872  0.04137618 0.04136582 0.04058741 0.04072956
 0.04137618 0.04154041 0.04065567 0.0393872  0.04135889 0.04069183
 0.04137618 0.04137618]
[0.04137344 0.04137344 0.04137344 0.04137344 0.04141165 0.03841165
 0.04281699 0.0393846  0.04137344 0.04136309 0.04058473 0.04072687
 0.04137344 0.04153766 0.04065299 0.0393846  0.04135615 0.04068915
 0.04137344 0.04137344]
[0.04056563 0.04056563 0.04056563 0.04056563 0.04060309 0.03766167
 0.04198099 0.03861562 0.04056563 0.04055547 0.03979232 0.03993168
 0.04056563 0.04072664 0.03985924 0.03861562 0.04054868 0.03989469
 0.04056563 0.04056563]
[0.03936305 0.03936305 0.03936305 0.03936305 0.03939939 0.03654517
 0.04073645 0.03747084 0.03936305 0.03935319 0.03861266 0.03874789
 0.03936305 0.03951928 0.0386776  0.03747084 0.0393466  0.038712
 0.03936305 0.03936305]
[0.04137618 0.04137618 0.04137618 0.04137618 0.04141438 0.03841419
 0.04281982 0.0393872  0.04137618 0.04136582 0.04058741 0.04072956
 0.04137618 0.04154041 0.04065567 0.0393872  0.04135889 0.04069183
 0.04137618 0.04137618]
[0.04146411 0.04146411 0.04146411 0.04146411 0.0415024  0.03849583
 0.04291082 0.03947091 0.04146411 0.04145373 0.04067367 0.04081613
 0.04146411 0.04162869 0.04074208 0.03947091 0.04144679 0.04077831
 0.04146411 0.04146411]
[0.0421028  0.0421028  0.0421028  0.0421028  0.04214168 0.0390888
 0.0435718  0.0400789  0.0421028  0.04209226 0.04130019 0.04144483
 0.0421028  0.04226992 0.04136965 0.0400789  0.04208521 0.04140644
 0.0421028  0.0421028 ]
[0.0393872  0.0393872  0.0393872  0.0393872  0.03942357 0.0365676
 0.04076144 0.03749383 0.0393872  0.03937734 0.03863635 0.03877167
 0.0393872  0.03954354 0.03870133 0.03749383 0.03937074 0.03873575
 0.0393872  0.0393872 ]
[0.04137121 0.04137121 0.04137121 0.04137121 0.04140941 0.03840958
 0.04281468 0.03938247 0.04137121 0.04136085 0.04058254 0.04072468
 0.04137121 0.04153542 0.0406508  0.03938247 0.04135392 0.04068695
 0.04137121 0.04137121]
[0.03925007 0.03925007 0.03925007 0.03925007 0.03928631 0.03644028
 0.04061953 0.03736329 0.03925007 0.03924024 0.03850183 0.03863668
 0.03925007 0.03940586 0.03856659 0.03736329 0.03923366 0.03860089
 0.03925007 0.03925007]
[0.04137618 0.04137618 0.04137618 0.04137618 0.04141438 0.03841419
 0.04281982 0.0393872  0.04137618 0.04136582 0.04058741 0.04072956
 0.04137618 0.04154041 0.04065567 0.0393872  0.04135889 0.04069183
 0.04137618 0.04137618]
[0.04137618 0.04137618 0.04137618 0.04137618 0.04141438 0.03841419
 0.04281982 0.0393872  0.04137618 0.04136582 0.04058741 0.04072956
 0.04137618 0.04154041 0.04065567 0.0393872  0.04135889 0.04069183
 0.04137618 0.04137618]]
```

[12]:
```
display(' since all entries are approximately equal to 0.05, then we can prove␣
 ↪the matrix A is reversible, and the pi[0.05,0.05,....,0.05]is the initial␣
 ↪distribution')
display(' since all entries are greater than 0 after repeation(10,100␣
 ↪times),then we can prove the matrix A is irreducible')
```

```
' since all entries are approximately equal to 0.05, then we can prove the matrix A is reversibl

' since all entries are greater than 0 after repeation(10,100 times),then we can prove the matri
```

[14]:
```
#A_2 is new transition unver metropolis algorithm


def travel_time(i):
    travel_1 = 0
    # start from 1 come back to 1
    #for i in range(0, 20, 1):
    for j in range(0, 20, 1):
            #1 transit point  i -> j  -> i
            travel_1 += A_2[i][j] * Q_2[i][j] + A_2[j][i] * Q_2[j][i] # distance␣
 ↪* probability

    return travel_1


def travel_time_2(i):
    distance_2 = 0
    for j in range(0, 20, 1):
             for k in range(0, 20, 1):
                # 2 transit point  i -> j -> k -> i
                    distance_2 += A_2[i][j] * Q_2[i][j] + A_2[j][k] * Q_2[j][k]␣
 ↪+ A_2[k][i] * Q_2[k][i] # distance * probability
    return distance_2

for i in range(0, 20, 1):
    print("the 1 iteration average time of city ", i," is ",travel_time(i) )
    print("the 2 iteration average time of city ", i," is ",travel_time_2(i) )
    print(travel_time)
```

```
the 1 iteration average time of city  0  is  72.6446240748816
the 2 iteration average time of city  0  is  2272.7862271537856
<function travel_time at 0x0000028421DA9280>
the 1 iteration average time of city  1  is  95.35328032721448
the 2 iteration average time of city  1  is  2726.9593522004375
```

```
<function travel_time at 0x0000028421DA9280>
the 1 iteration average time of city  2  is  68.2588816778353
the 2 iteration average time of city  2  is  2185.071379212859
<function travel_time at 0x0000028421DA9280>
the 1 iteration average time of city  3  is  76.31661958260571
the 2 iteration average time of city  3  is  2346.2261373082656
<function travel_time at 0x0000028421DA9280>
the 1 iteration average time of city  4  is  75.78655121649018
the 2 iteration average time of city  4  is  2335.6247699859578
<function travel_time at 0x0000028421DA9280>
the 1 iteration average time of city  5  is  71.23964828655951
the 2 iteration average time of city  5  is  2244.686711387343
<function travel_time at 0x0000028421DA9280>
the 1 iteration average time of city  6  is  97.48348762776247
the 2 iteration average time of city  6  is  2769.5634982114016
<function travel_time at 0x0000028421DA9280>
the 1 iteration average time of city  7  is  68.70277319958304
the 2 iteration average time of city  7  is  2193.9492096478143
<function travel_time at 0x0000028421DA9280>
the 1 iteration average time of city  8  is  80.99742037612431
the 2 iteration average time of city  8  is  2439.842153178639
<function travel_time at 0x0000028421DA9280>
the 1 iteration average time of city  9  is  88.6018916588495
the 2 iteration average time of city  9  is  2591.931578833141
<function travel_time at 0x0000028421DA9280>
the 1 iteration average time of city  10  is  88.21090683802635
the 2 iteration average time of city  10  is  2584.111882416681
<function travel_time at 0x0000028421DA9280>
the 1 iteration average time of city  11  is  96.03681534846473
the 2 iteration average time of city  11  is  2740.6300526254468
<function travel_time at 0x0000028421DA9280>
the 1 iteration average time of city  12  is  99.31336885362722
the 2 iteration average time of city  12  is  2806.1611227287003
<function travel_time at 0x0000028421DA9280>
the 1 iteration average time of city  13  is  89.27726524823913
the 2 iteration average time of city  13  is  2605.4390506209356
<function travel_time at 0x0000028421DA9280>
the 1 iteration average time of city  14  is  78.85350532408285
the 2 iteration average time of city  14  is  2396.9638521378106
<function travel_time at 0x0000028421DA9280>
the 1 iteration average time of city  15  is  68.41528893955049
the 2 iteration average time of city  15  is  2188.1995244471645
<function travel_time at 0x0000028421DA9280>
the 1 iteration average time of city  16  is  78.26742924471891
the 2 iteration average time of city  16  is  2385.24233055053
<function travel_time at 0x0000028421DA9280>
the 1 iteration average time of city  17  is  89.74616552786652
the 2 iteration average time of city  17  is  2614.817056213483
```

```
<function travel_time at 0x0000028421DA9280>
the 1 iteration average time of city  18  is  75.36881176536473
the 2 iteration average time of city  18  is  2327.2699809634473
<function travel_time at 0x0000028421DA9280>
the 1 iteration average time of city  19  is  80.9127561944581
the 2 iteration average time of city  19  is  2438.1488695453145
<function travel_time at 0x0000028421DA9280>
```

[9]:
```python
def check_symmetric(a, rtol=1e-05, atol=1e-08): # check Q is symmetry
    return not np.allclose(a, a.T, rtol=rtol, atol=atol)

check_symmetric(A)
print("Check new Transition Matrix A is symmetry: ",check_symmetric(A))
display(We have prove this  travelling matrix is symmetry )
```

```
Check new Transition Matrix A is symmetry:  True
```

[ ]:
```python

```