Name : Parigya Jain
Roll No. : 2K18/MC/077

PROGRAM 9

AIM :

Demonstrating Markov Chain (Cont.) WAP to implement Markov Chain special cases

(a) To find steady state probabilities in case of ergodic Markov Chain

(b) To find that the specific state in a Markov chain is a recurrent or transient

THEORY :

A Markov chain is a stochastic model describing a sequence of possible events in which the probability of each event depends only on the state attained in the previous event. A countably infinite sequence, in which the chain moves state at discrete time steps, gives a discrete-time Markov chain.
A Markov chain is called an ergodic or irreducible Markov chain if it is possible to eventually get from every state to every other state with positive probability.

RESULT :

(a) Eg: A traveller visits 4 cities A, B, C, D if he visits A then he is equally likely to visit B, C but not D. If he visits B then he is twice as likely to go to C than A or D . If he visits C then he is 2 times as likely to go to A than B but he will not go to D. If he visits D then he is equally likely to go to A, B, C.

CODE :

```
 function [ answer ] = ergodic(tpm,n)
 tpm = -tpm;
 for i=1:n
 for j=1:n
if i==j
 tpm(i,j) = 1;
 end
 end
 end

A = [tpm';ones(1,n)];
```

Name : Parigya Jain
Roll No. : 2K18/MC/077

```
B = zeros(n+1,1);
B(n+1) = 1;
answer = linsolve(A,B);
end
```

OUTPUT :

```
>> t = [0 1/2 1/2 0;1/4 0 1/2 1/4;2/3 1/3 0 0;1/3 1/3 1/3 0]
t =

0         0.5000    0.5000    0

0.2500    0         0.5000    0.2500

0.6667    0.3333    0         0

0.3333    0.3333    0.3333    0

>> ans = ergodic(t,4)
ans =

0.3133

0.2892

0.3253

0.0723
```

Name : Parigya Jain
Roll No. : 2K18/MC/077

(b) Eg: A traveller visits 4 cities A, B, C, D if he visits A then he is equally likely to visit B, C but not D. If he visits B then he is twice as likely to go to C than A or D. If he visits C then he is 2 times as likely to go to A than B but he will not go to D. If he visits D then he is equally likely to go to A, B, C.

```
In [1]:  import numpy as np
         import math

In [3]:  a= np.array([[1, 0 , 0 , 0, 0], [0 , 1, 0 ,0, 0] , [0 ,0 ,0.5, 0.5 ,0], [0 , 0 , 0.5, 0.5, 0] , [0.25, 0.25, 0 , 0, 0.5]])

In [5]:  transient=[]
         recurrent=[]
         row, col=a.shape
         a

Out[5]:  array([[1.  , 0.  , 0.  , 0.  , 0.  ],
                [0.  , 1.  , 0.  , 0.  , 0.  ],
                [0.  , 0.  , 0.5 , 0.5 , 0.  ],
                [0.  , 0.  , 0.5 , 0.5 , 0.  ],
                [0.25, 0.25, 0.  , 0.  , 0.5 ]])

In [6]:  for i in range(row):
             flag=True
             for j in range(col) :
                 if a[i][j]>0:
                     if a[j][i]==0 :
                         flag=False
                         transient.append(i)
                         break
             if flag :
                 recurrent.append(i)

In [7]:  transient

Out[7]:  [4]

In [8]:  recurrent
Out[8]:  [0, 1, 2, 3]
```

DISCUSSION :

We successfully found the steady state probabilities in case of ergodic Markov Chain and found if the given Markov Chain is recurrent or transient.