Name: Meet Popat
Roll No. : 2020320
SPA Assignment-2

Ques1: Discrete Chain Markov Chain

State Space = {1,2}

a. The chain starts in the first state with probability 1.
   Then we generate the next state with a random function in R and store it states.
   We determine the Random Variable in each iteration and then we plot it using matplot.

   The graph shows the states 1 and 2 over 50 times i.e 50 times and at each time in which state we are.

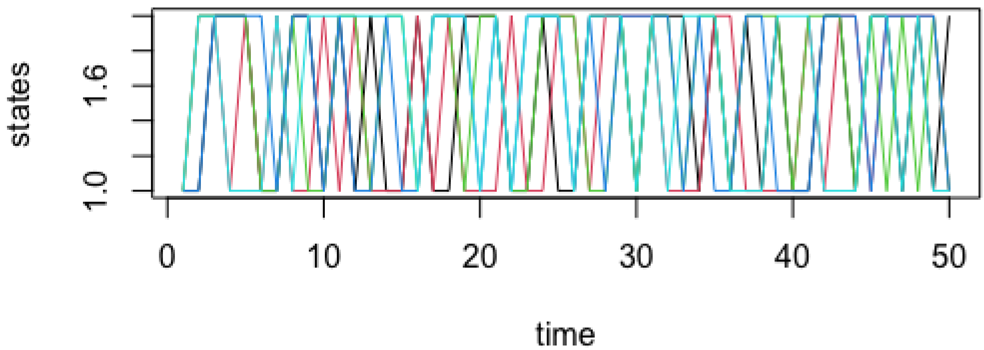
   The graph shows the simulation for 5 times in different colors i.e green, blue, red, black and cyan.

   <u>My code for plot in R:</u>

```
P <- matrix(c(0.3,0.5,0.7,0.5),nrow=2,ncol=2)
chain.states <- matrix(ncol = 5, nrow = 50)
for(i in seq_len(5)){
  states <- 2 #states
  states <- 50
  states[1] <- 1

  for(x in 2:50) {
    p  <- P[states[x-1], ]
    states[x] <- which(rmultinom(1, 1, p) == 1)
  }
  chain.states[,i] <- states
}
matplot(chain.states, type='l', lty=1, col=1:5,  ylab='states', xlab='time')
```

Plot:



b. We run a loop in R to multiply and find power of the matrix P in R.
We find the Values for P10, P20 and P50 and observe that all the 3 have the same values this is because the Markov chain is irreducible i.e every state is accessible from every other state and aperiodic i.e all the states have period 1.
This implies that the Limiting and Stationary distribution of the Markov Chain is the same.

Hence, The Markov Chain reaches the limiting distribution on or before P10 and hence we get the above further power of the matrix to be the same.

The code for the simulation is :

```
ans <- matrix(c(1,0,0,1),nrow = 2, ncol = 2)
P <- matrix(c(0.3,0.5,0.7,0.5),nrow=2,ncol=2)

for (i in 1:51) {
  ans <- ans %*% P   #multiply
  if(i == 10) {
    print("P^10")
    print(ans)
  }
  if(i == 20) {
    print("P^20")
    print(ans)
  }
  if(i == 50) {
    print("P^50")
    print(ans)
  }
}
```

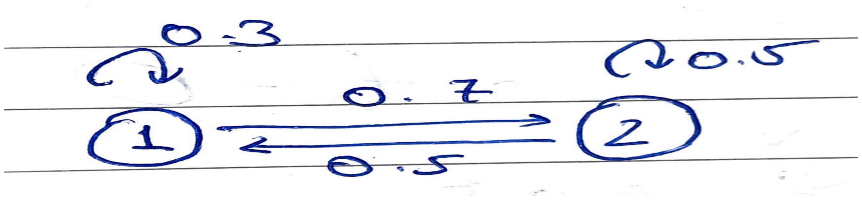
The values of The power of matrices:

```
[1] "P^10"
          [,1]      [,2]
[1,] 0.4166667 0.5833333
[2,] 0.4166666 0.5833334
[1] "P^20"
          [,1]      [,2]
[1,] 0.4166667 0.5833333
[2,] 0.4166667 0.5833333
[1] "P^50"
          [,1]      [,2]
[1,] 0.4166667 0.5833333
[2,] 0.4166667 0.5833333
```

<u>Observation:</u>

We find the Values for P10, P20 and P50 and observe that all the 3 have the same values this is because the Markov chain is irreducible i.e every state is accessible from every other state and aperiodic i.e all the states have period 1.
This implies that the Limiting and Stationary distribution of the Markov Chain is the same.
The period for state 1 is 1 and same for state 2 is also 1 and hence aperiodic irreducible Markov's chain.

The Markov chain is:



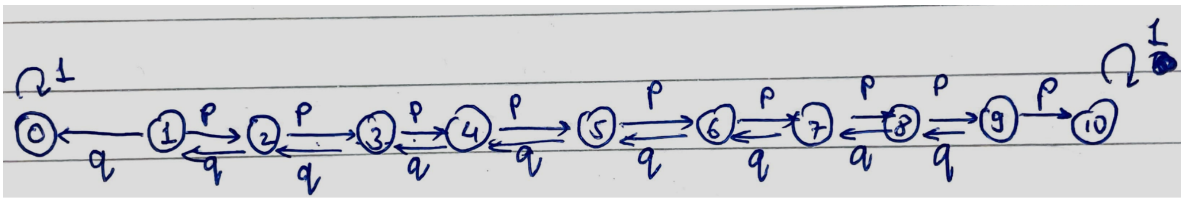
As in stationary distribution $\Pi=\Pi P$ and summation ,Πj where J belongs to S is 1 we get all the three i.e P^10, P^20 and P^50 as same.

Ques 2: Gambler’s Ruin Problem

a. The winning probability for A is 0.8 and losing probability is 0.2
b. The winning probability for B is 0.2 and losing probability is 0.8

They both start with 5 dollars and want to reach the target of 10 dollars and they go bankrupt when they will have no dollars i.e 0 dollars.
When they win they will win 1 dollar at each win and lose 1 dollar when they lose the bet.

The random walk will be :



Hence to simulate a Random walk for both A and B, we will run a loop till the balance becomes zero or till it reaches 10, We we generate 1 and -1 with winning and losing probability respectively and if we win we will add 1 dollars to the current balance and subtract 1 dollar if we lose till we reach our target state.

<u>The code for the simulation 1D random walk is:</u>

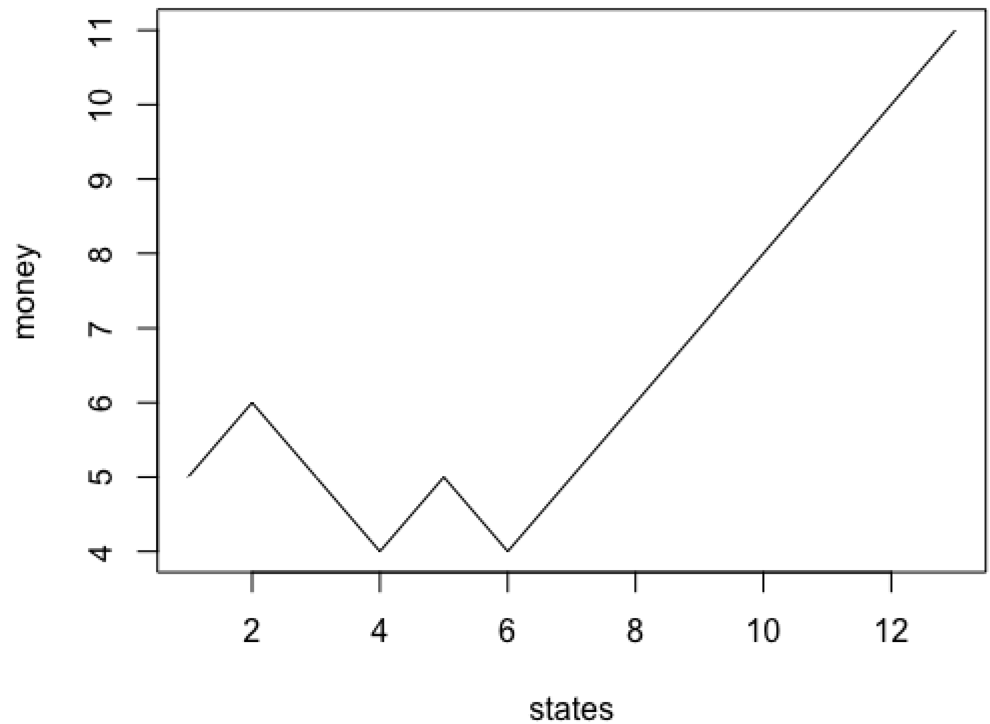
```
start_balance <- 5
bet <- 1
current_balance <- start_balance
vec = as.vector(0)
vec[1]= start_balance

while(current_balance>0 && current_balance<10){ #till he hits the target or goes bankruptcy
  outcome <- sample(c(-1,1), size=1, replace=TRUE,prob=c(0.2,0.8)) #generates a random walk
  if(outcome == 1){ #if he win add the bet i.e 1
    current_balance = current_balance + bet

  }
  else{ #else remove 1
    current_balance = current_balance - bet
  }
  vec <- append(vec,current_balance)

}
plot(seq(1,length(vec)),vec,xlab='states',ylab='money',type='l')
```

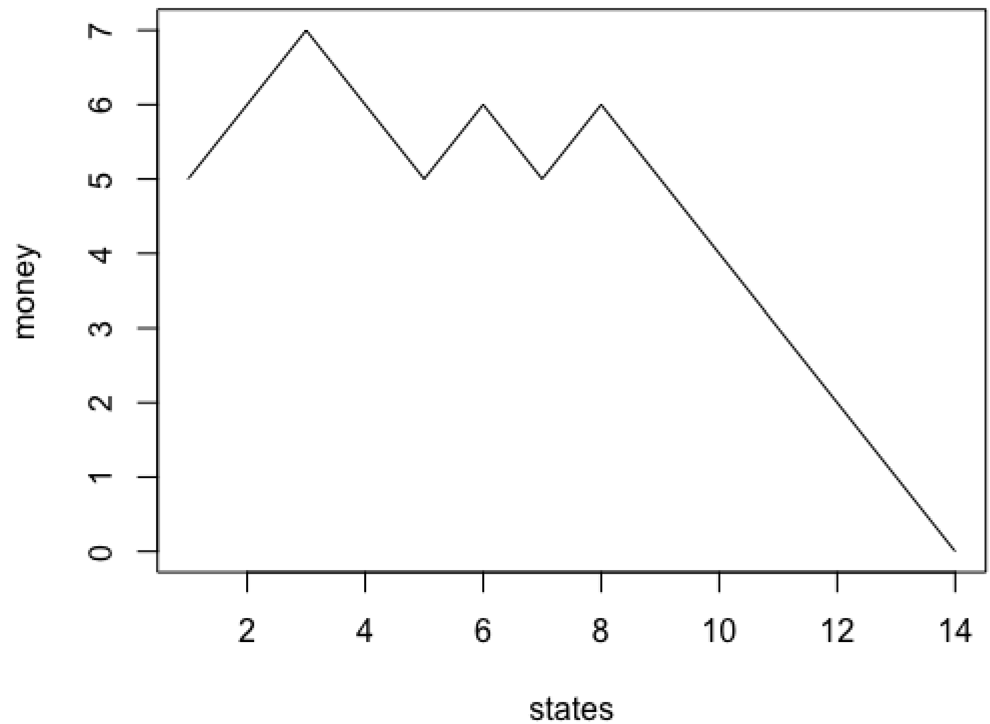
The plots are A is :



As A has a winning probability greater than 0.5 he can win and reach the target of 10 dollars in a random amount of states/time as we generate different random walks.

The state space of this Markov chain is:
S= {0,1,2,3,4,5,6,7,8,9,10}
And all the assumptions of Markov's chain are considered to be met in the question.

The plot for B is:



As B has a winning probability of less than 0.5 it is sure to go bankrupt and reach 0 dollars in a random amount of states/time as we generate different random walks.

The state space of this Markov chain is:
S= {0,1,2,3,4,5,6,7,8,9,10}
And all the assumptions of Markov's chain are considered to be met in the question.