INFX 576: Problem Set 7 - Network Dynamics*

Avanti Chande

Due: Thursday, March 2, 2017

Collaborators:

Instructions:

Before beginning this assignment, please ensure you have access to R and RStudio.

1. Download the `problemset7.Rmd` file from Canvas.
2. Replace the "Insert Your Name Here" text in the `author:` field with your own full name. Any collaborators must be listed on the top of your assignment.
3. Be sure to include well-documented (e.g. commented) code chucks, figures and clearly written text chunk explanations as necessary. Any figures should be clearly labeled and appropriately referenced within the text.
4. Collaboration on problem sets is acceptable, and even encouraged, but each student must turn in an individual write-up in his or her own words and his or her own work. The names of all collaborators must be listed on each assignment. Do not copy-and-paste from other students' responses or code.
5. When you have completed the assignment and have **checked** that your code both runs in the Console and knits correctly when you click `Knit PDF`, rename the R Markdown file to `YourLastName_YourFirstName_ps7.Rmd`, knit a PDF and submit the PDF file on Canvas.

Setup:

In this problem set you will need, at minimum, the following R packages.

```
# Load standard libraries
library(statnet)
```

```
## Warning: package 'statnet' was built under R version 3.2.5
```

```
## Warning: package 'tergm' was built under R version 3.2.5
```

```
## Warning: package 'statnet.common' was built under R version 3.2.5
```

```
## Warning: package 'ergm' was built under R version 3.2.5
```

```
## Warning: package 'network' was built under R version 3.2.5
```

```
## Warning: package 'networkDynamic' was built under R version 3.2.5
```

```
## Warning: package 'ergm.count' was built under R version 3.2.5
```

```
## Warning: package 'sna' was built under R version 3.2.5
```

```
load("block.fit.Rdata")
library(degreenet)
```

```
## Warning: package 'degreenet' was built under R version 3.2.5
```

*Problems originally written by C.T. Butts (2009)

Problem 1: Social Processes and Structure

Imagine a social group comprised of individuals from two cultures: the "reds" and the "blues." Both groups are alike in their desire for transitive friendships; however, they differ in how they react to intransitivity.

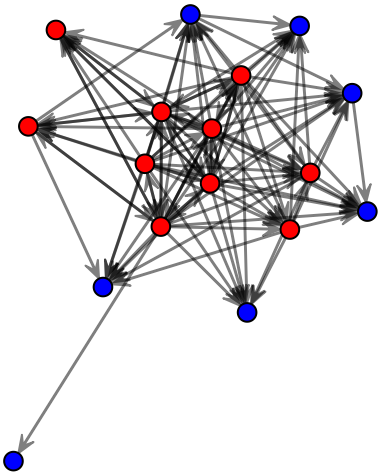
When placed in an arbitrary network, reds tend to act by extending friendship towards those with whom they have a transitive precondition. In other words, the condition $v_i \to v_j$ and $v_j \to v_k$, generates $v_i \to v_k$. Blues, in contrast, generally respond by severing ties which are associated with uncompleted transitive states. In other words, $v_i \to v_j \to v_k$, leads to $v_i \not\to v_j$.

To simulate the above process on a network with `nred` "red" and `nblue` "blue" vertices, use the following function:

```
transsim<-function(nred,nblue,action.prob.red=c(0.9,0.05,0.05),
                   action.prob.blue=c(0.05,0.9,0.05),initial.density=0.2,
                   plot.result=TRUE,return.graph=FALSE,...){
  n<-nred+nblue
  #Draw the initial graph
  g<-rgraph(n,tp=initial.density)
  rownames(g)<-c(rep("r",nred),rep("b",nblue))
  colnames(g)<-c(rep("r",nred),rep("b",nblue))
  #Now, update in a random order
  uo<-sample(1:n)
  for(i in uo){
    for(j in 1:n)
      for(k in 1:n)
        if((i!=j)&&(j!=k)&&(i!=k)){
          if((g[i,j]*g[j,k])&&!g[i,k]){
            if(i<=nred)
              action<-sample(1:3,1,prob=action.prob.red)
            else
              action<-sample(1:3,1,prob=action.prob.blue)
            if(action==1)
              g[i,k]<-1
            else if(action==2)
              g[i,j]<-0
          }
        }
  }
  #Plot the graph, if necessary
  if(plot.result)
    gplot(g,vertex.col=c(rep(2,nred),rep(4,nblue)),edge.col=rgb(0,0,0,.5))
  #Return, if needed
  if(return.graph)
    return(g)
  invisible()
}
```

The above function should produce a plot of the resulting network. You can return the adjacency matrix by setting the `return.graph` argument to be `TRUE`. The first N vertices will be the red ones, followed by the blue ones.

```
transsim(10,7,action.prob.red=c(0.9,0.05,0.05),
                   action.prob.blue=c(0.05,0.9,0.05),initial.density=0.2,
                   plot.result=TRUE,return.graph=TRUE)
```

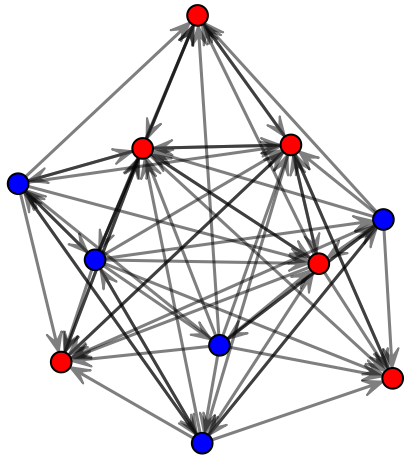


```
##   r r r r r r r r r r b b b b b b b
## r 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
## r 0 0 1 1 0 1 1 1 1 1 1 1 1 0 0 1 1
## r 0 0 0 0 0 0 0 0 0 0 1 1 1 1 0 1 1
## r 0 0 1 0 0 0 0 0 0 0 1 1 1 1 0 1 1
## r 1 1 1 1 0 1 1 1 1 1 1 1 1 1 0 1 1
## r 1 1 1 1 0 0 1 1 1 1 1 1 1 1 0 1 1
## r 1 1 1 1 0 1 0 1 1 1 1 1 1 1 0 1 1
## r 1 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0
## r 1 1 1 1 0 1 1 1 0 1 1 1 1 1 0 1 1
## r 1 0 0 0 0 0 1 0 0 0 0 1 0 1 0 0 0
## b 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
## b 0 0 0 0 0 0 0 0 0 0 1 0 1 0 0 0 0
## b 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1
## b 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
## b 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
## b 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
## b 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

You can change the probabilities of adding an edge, dropping an edge, or taking no action (respectively) in response to intransitivity for each group with the `action.prob.red` and `action.prob.blue` arguments. For example setting `action.prob.red=c(0.9, 0.05, 0.05)` will set reds to response to intransitivity by adding an edge 90% of the time, dropping an edge 5% of the time, and ignoring the violation 5% of the time. Notice this is the default setting.

```
transsim(6,5,action.prob.red=c(0.5,0.6,0.02),
                    action.prob.blue=c(0.8,0.07,0.7),initial.density=0.5,
```

```
                    plot.result=TRUE,return.graph=TRUE)
```



```
##   r r r r r r b b b b b
## r 0 1 1 1 1 1 0 0 1 0 0
## r 1 0 1 1 1 1 1 0 0 1 0
## r 1 1 0 0 0 0 0 0 0 0 0
## r 1 1 0 0 0 0 0 0 0 0 0
## r 1 0 0 0 0 0 0 0 0 0 0
## r 1 1 1 1 1 0 0 0 0 0 0
## b 1 1 1 1 1 1 0 1 1 1 1
## b 1 1 1 1 1 1 0 0 1 0 1
## b 0 1 1 0 1 1 1 1 0 1 0
## b 1 1 1 1 0 1 1 0 1 0 0
## b 1 1 1 1 1 1 0 1 1 1 0
```

`initial.density` is the initial density of the network. `nred` and `nblue` give the number of nodes in each group.

(a) Prediction

Consider what might happen when persons from both groups are placed together in a single friendship network. Make an initial guess at what you think will happen.

When a person from the blue group initiates friendship with any of the red/blue members, and if this friend initiates another tie with any other person, the blue person severes its ties with this friend.

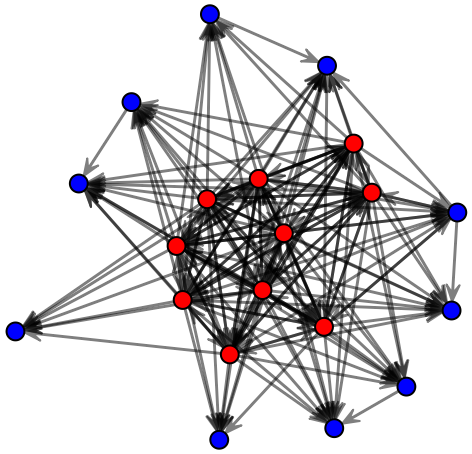
On the other hand, if a friendship is initiated by a red person, the relationship tends to be transitive

irrespective of the opposite friend being from a blue/red group.

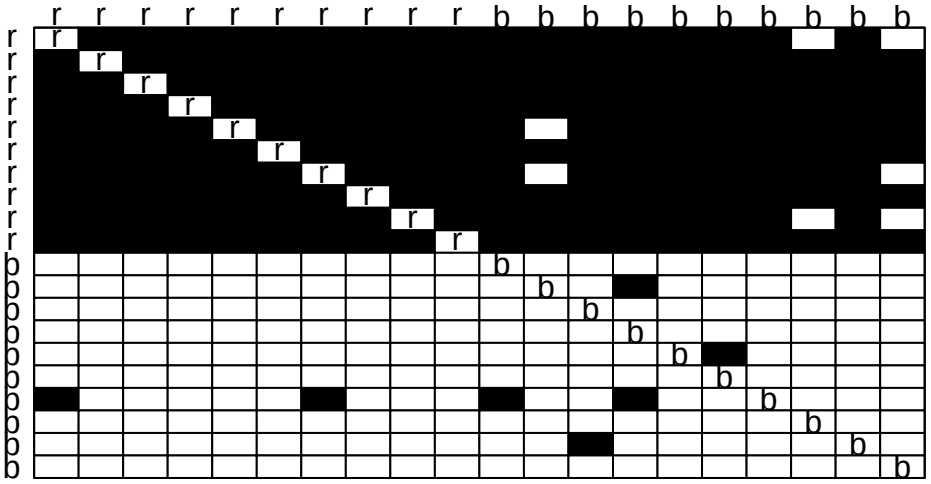
(a) Simulation

Design a simple simulation study using the `transsim` function. Consider a group of 10 red nodes and 10 blue nodes. After your experiment you should answer the following questions:

```
#Calling the transsim function
sim_1 <- transsim(10,10,action.prob.red=c(0.9,0.05,0.05),
                  action.prob.blue=c(0.05,0.9,0.05),initial.density=0.2,
                  plot.result=TRUE,return.graph=TRUE)
```



```
plot.sociomatrix(sim_1)
```



- What can you conclude about the structural effects of responses to intransitivity on the red and blue nodes?

The responses to intransitivity show that the blue nodes have a very low outdegree. They tend to severe ties in a intransitive pre-condition. That is why, friendship ties from blue to any other person are few. On the other hand, reds being responsive to transitive precondition, the structure of ties from red to blue and even red to red are in abundance.

- How is the total structure affected?

The combination of red and blue nodes in a network makes the network sparse. This is because blue tends to severe ties with their possessive behavior for their friends. The network would have been much more dense if red and blue nodes behaved similarly and had a similar reaction to a transitive pre-condition.

- How might you describe the overall state which tends to result from the combination of red and blue behaviors?

It looks like there is a core-periphery structure formed in this network with red nodes at the core and blue nodes at the periphery.

- How are these conclusions similar to or different from your initial guess about the outcome of the social process?

Initially we concluded that if blue forms a friendship with a red/blue, and if that person initiates another friendship tie with another red/blue, then blue severes ties with them. We can see a similar pattern in the network, as blue has a very low outdegree, and red nodes have a high outdegree.

Problem 2: Network Diffusion

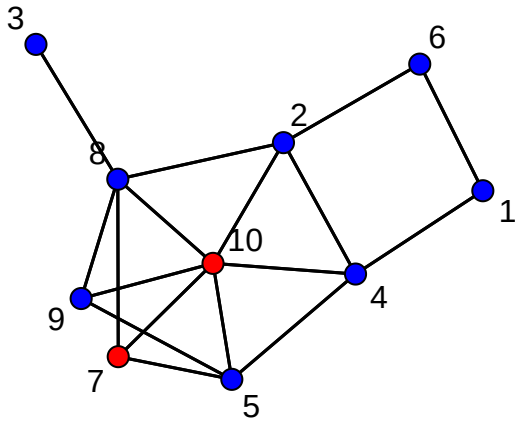
The following is a simple function to simulate cascading behavior in a social network. Write detailed comments for this code to demonstrate you understand the function. Feel free to modify or adjust the functionality.

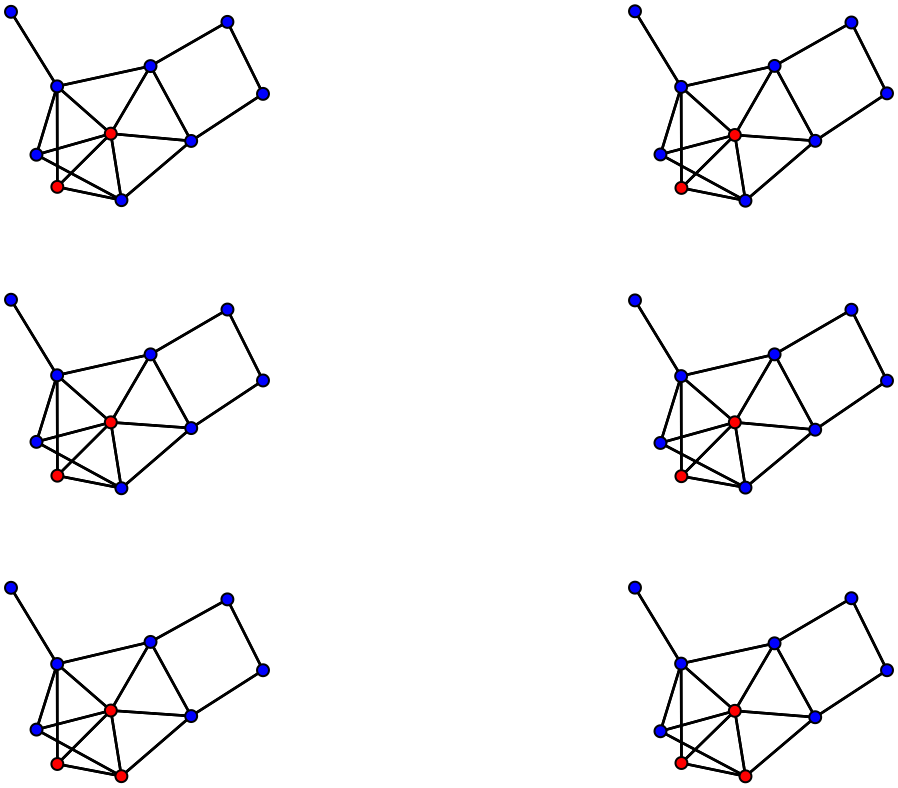
```
diffusion <- function(network_size=10, network_density=0.2, b_payout=2,
  a_payout=3, num_seeds=2, max_steps=10){

  #Generate a undirected random graph conditional on network size and density
  g <- network(rgraph(network_size, tprob=network_density), directed=FALSE)

  #Color the vertices blue
  vertex_colors <- rep("blue", network_size)

  #Randomly sample num_seeds (in this case, 2) nodes out of the nodes in the network size
  initial_nodes <- sample(1:network_size, num_seeds)

  #Color the randomly selected nodes red
  vertex_colors[initial_nodes] <- "red"

  #Plot the graph with lables and store the x,y co-ordinates in coords
  coords <- gplot(g, usearrows=FALSE, vertex.col=vertex_colors, displaylabel=TRUE)

  #Calculate q from initial adopters payoff to nodes interacting using behavior A
  q = b_payout / (b_payout + a_payout)

  #Divide the plot window
  par(mar=c(0,0,0,0), mfrow=c(3,2))

  #Calculate whether a particular node will switch to A

  step = 1

  #While step less than max_steps and all the nodes are not converted to red and the          #combina

  while(step < max_steps && sum(vertex_colors=="red") < network_size){
  for (i in 1:network_size){

  #Calculate the indices of each of the neighbors of each of the vertices in graph g
  neighborInds <- get.neighborhood(g, i, type="combined")

  #Calculate the number of neighbors that are red
  obsA <- sum(vertex_colors[neighborInds]=="red")

  #If the fraction of red nodes are greater than the threshold q, that particular vertex      color is

  if (obsA/length(neighborInds) >= q){
  vertex_colors[i] <- "red"
  }
  #Plot the new graph and repeat the process.
  gplot(g, usearrows=FALSE, vertex.col=vertex_colors, coord=coords)

  #Pause to examine the plot
  Sys.sleep(.2)
  step <- step + 1
  }
```

```
  }
}
```

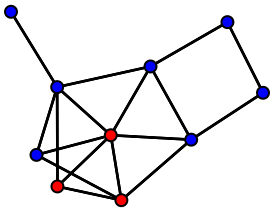
Experiment with this function to answer the following questions:

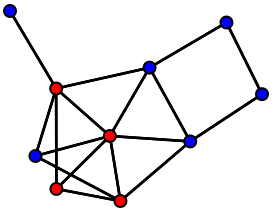
```
diffusion()
```

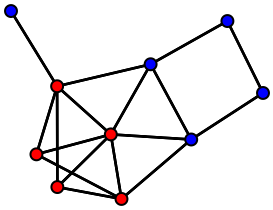


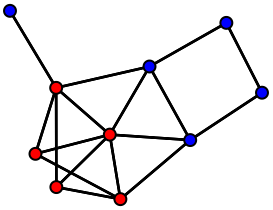


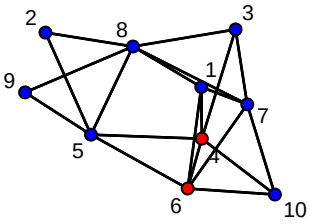
```
diffusion(max_steps = 30)
```

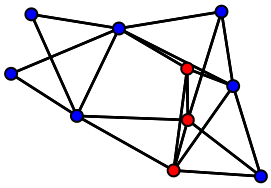


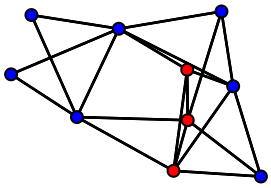


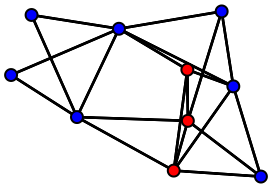


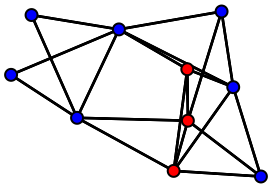


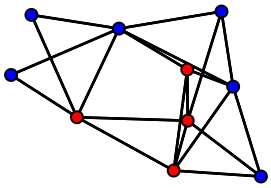

2
8
3
1
9
7
5
4
6
10

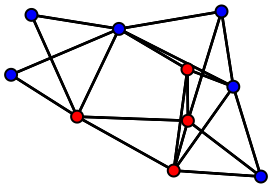


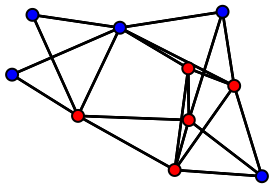


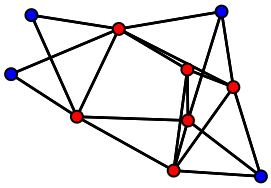


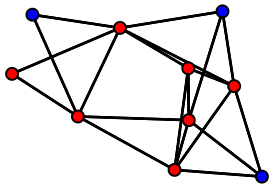


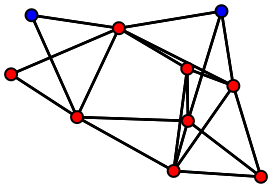


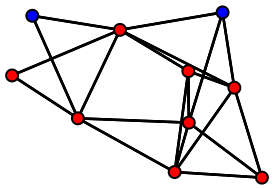


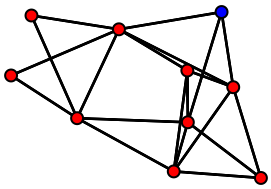


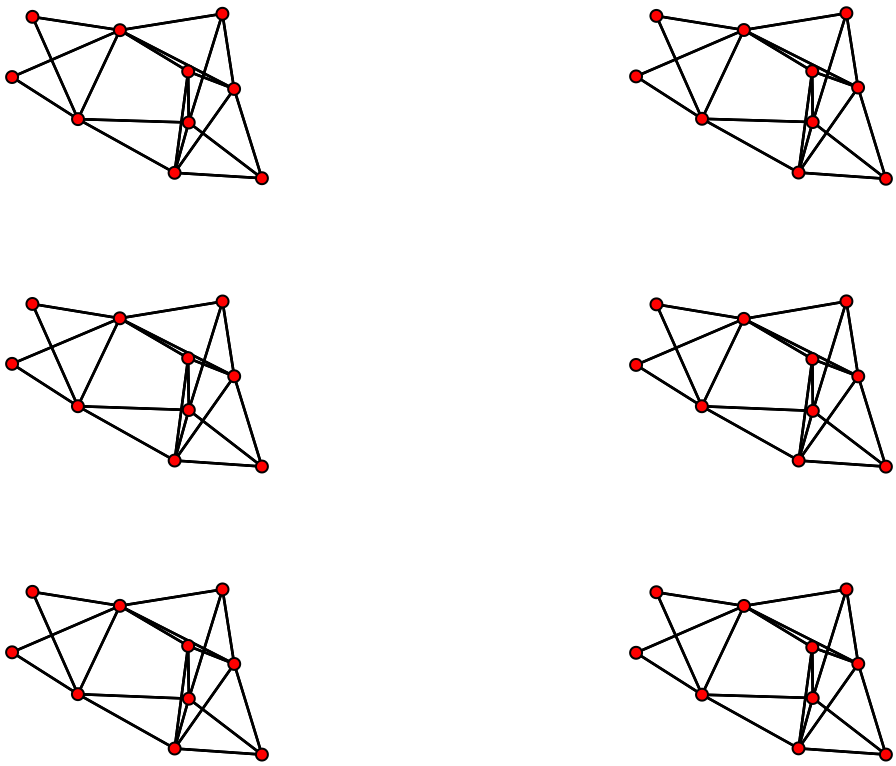




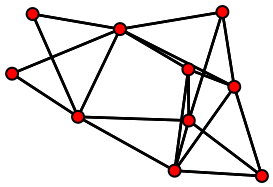


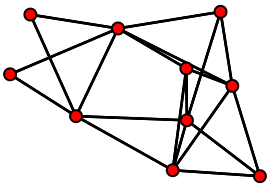


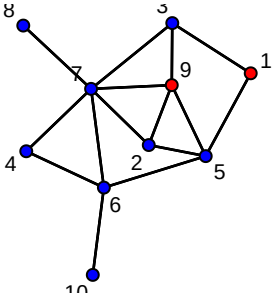


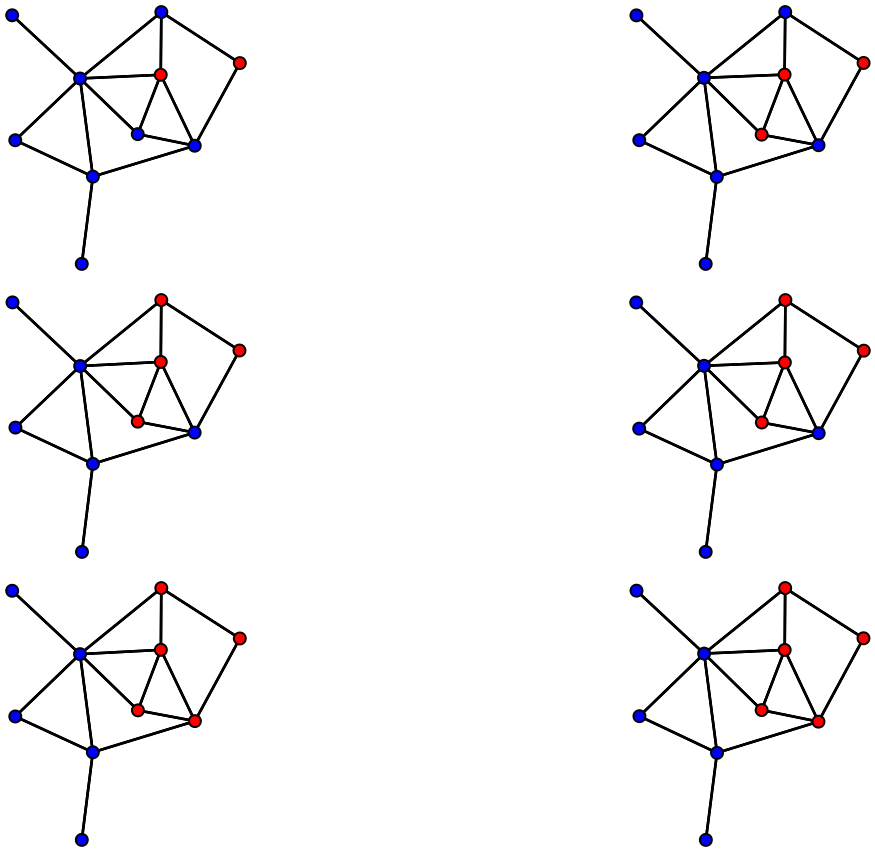


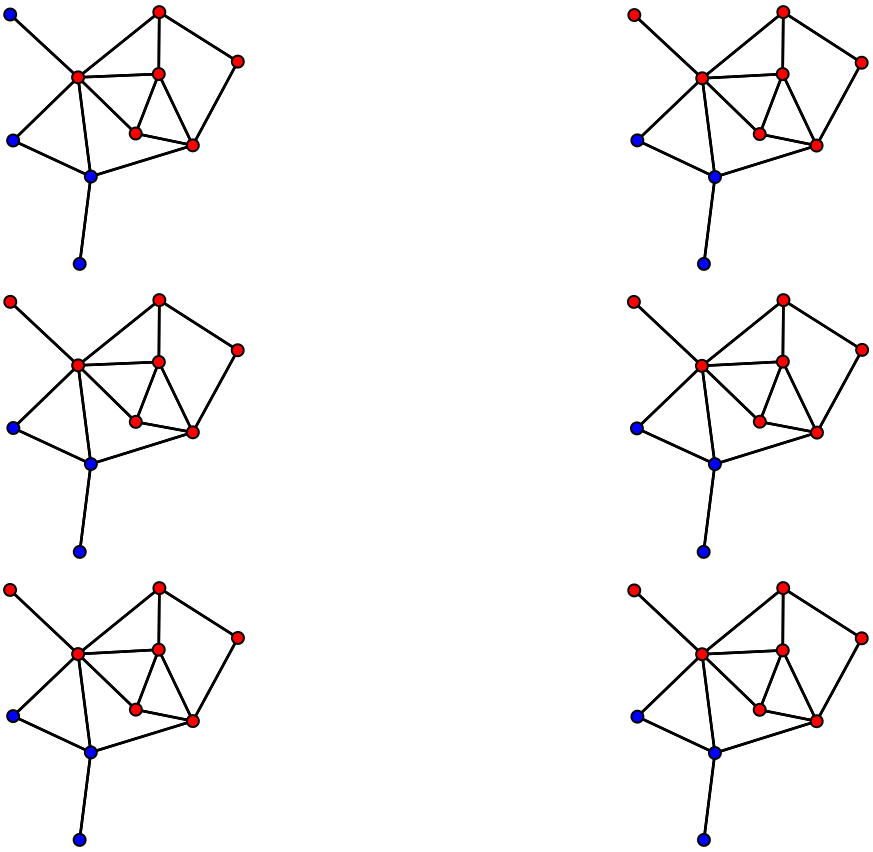
```
diffusion(network_size=10, network_density=0.2, b_payout=3,a_payout=7, num_seeds=2, max_steps=20)
```

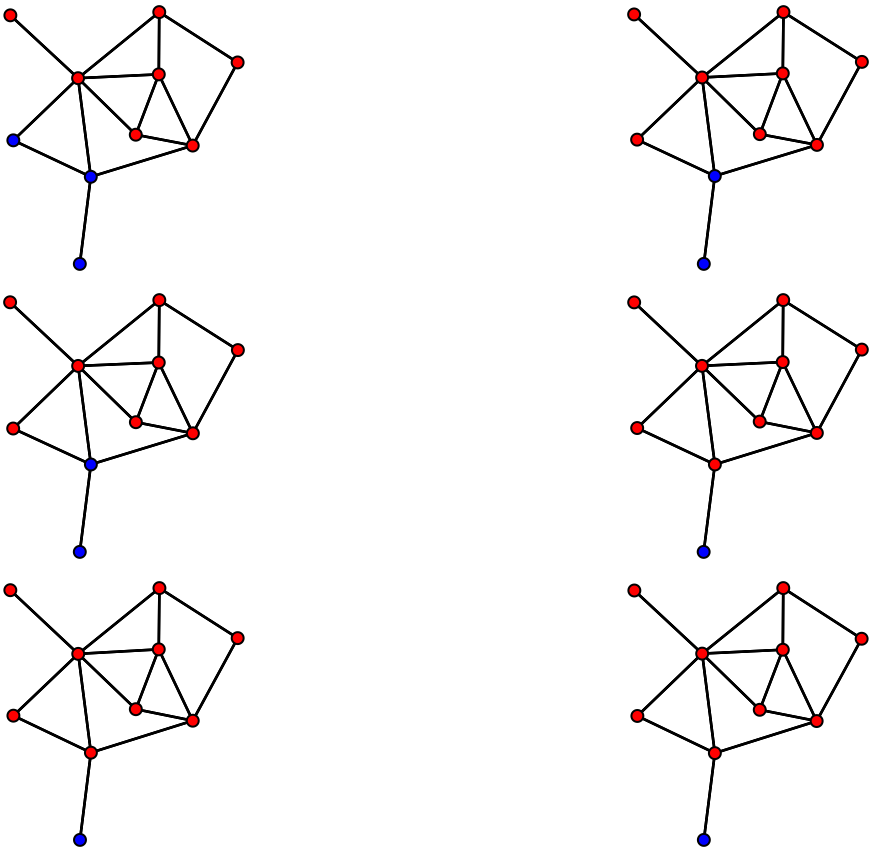




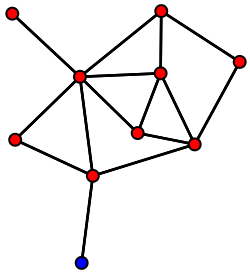

8
3
1
7
9
4
2
5
6
10

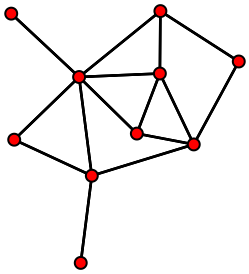


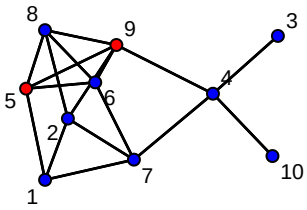


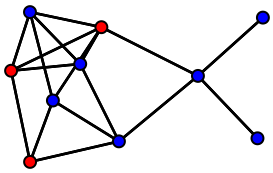


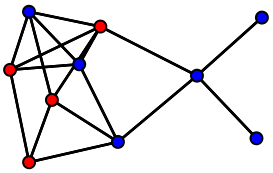
```
diffusion(network_size=10, network_density=0.2, b_payout=1,a_payout=9, num_seeds=2, max_steps=10)
```

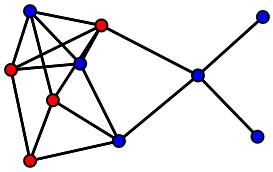


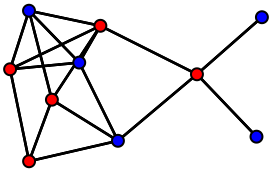


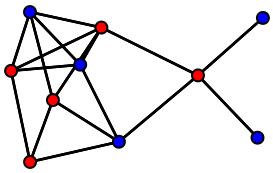

8
9
3
4
5
6
2
7
10
1

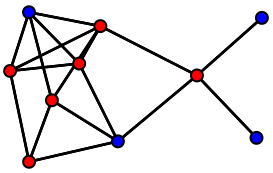


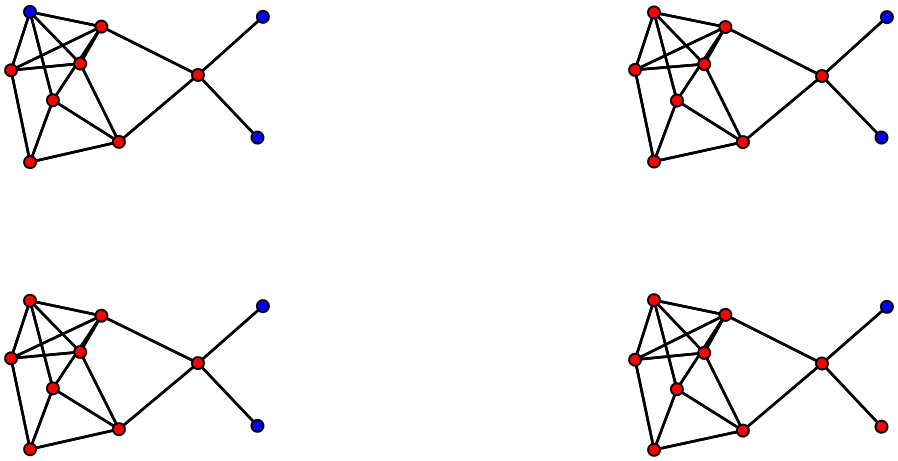












- When does this simulation result in everyone switching to A? The simulation may result in everyone switching to A in the following cases:

1. When the initial adopters are connected to a large number of nodes in the network.
2. It may also increase when the initial adopters are high.
3. When we increase the threhold value q.
4. when we increase the max_step size, there are chances that every node will be visited more than once, and hence, switch to A.

- What causes the spread of A to stop? The spread of A stops in the following cases:

1. When the threshold q is high
2. When the initial adopters are few and are located at the ends of the network. That is, when the initial adopters are not connected to many other nodes in the network.
3. When the max_steps is less than the network size.