Assignment: Cascades

By Riya Gupta (17BCS026)

Run the code.R file to execute function initializations and definitions.

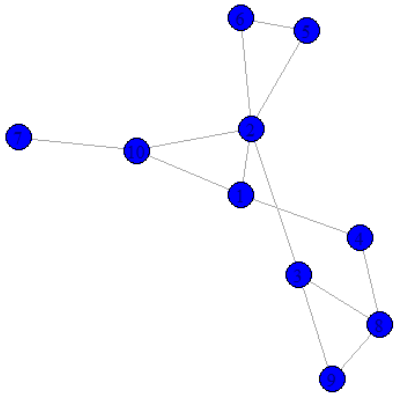
makeGraph() function takes user input for number of nodes and edges and inputs all the edges one by one to make and return a graph.

runprogram(graph) function is to visualize the cascading behavior of a graph and return the cascading size, i.e., the number of nodes that have adopted the new behavior.
This function asks for user inputs for threshold values a (new behavior) and b (old behavior), shown in red and blue colors, respectively. Then it asks for seed nodes and then plots a comparison between the initial graph with marked seeds and the graph after cascading.

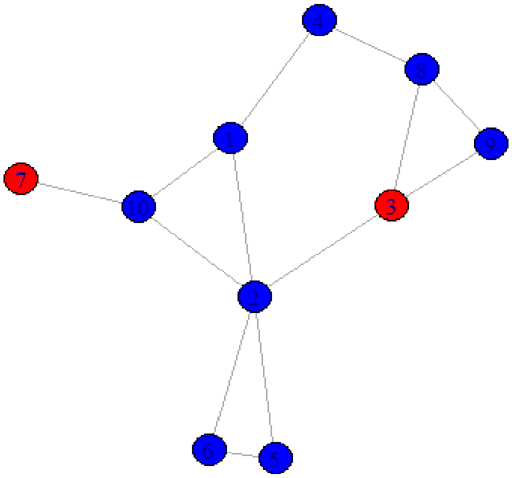
If all the nodes in a network adopts a new behavior because of the seed of a few nodes, it is said that a complete cascade occurred, whereas if only some nodes adopted the new behavior, it is said that it was as incomplete cascade.
The more the value of a/b, where a and b are the threshold values of behaviors A and B, respectively, the more is the chance of a complete cascade occurring.

It was seen that the position of seed nodes impacted the overall adoption of the new behavior, the spread was difficult in case of a highly connected cluster within a network, if a network had two communities joined by only a few edges then the presence of seeds in a community affects only that particular community and also if the seed nodes were far from each other in a network then too the cascading were rather dependent on a higher payoff value.

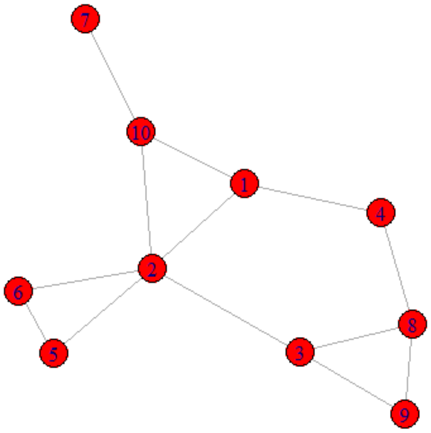
1. The following graph shows a network where every node followed behavior B.



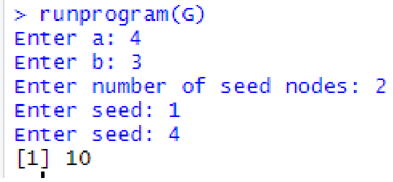
Nodes 3 and 7 were seeded with behavior A, hence are now marked in red color (as shown below).



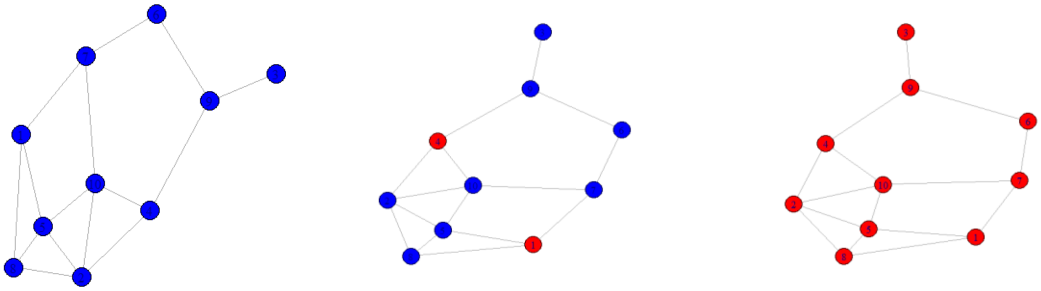
Putting a = 5 and b = 3, that every node adopted the new behavior and thus, cascading size = 10.



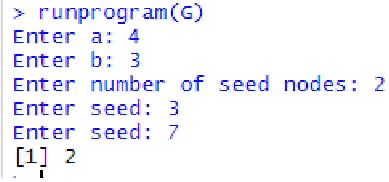
2. Similarly, for the following example there is a **complete cascade**, i.e., every node adopts the new behavior, where a=4, b=3, seeds are 1 and 4.



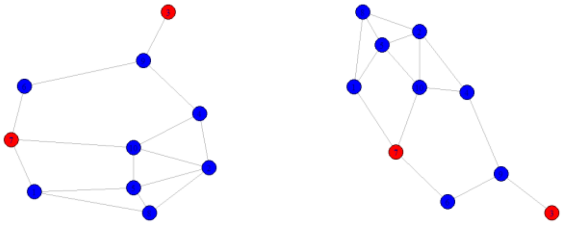
```
> runprogram(G)
Enter a: 4
Enter b: 3
Enter number of seed nodes: 2
Enter seed: 1
Enter seed: 4
[1] 10
```



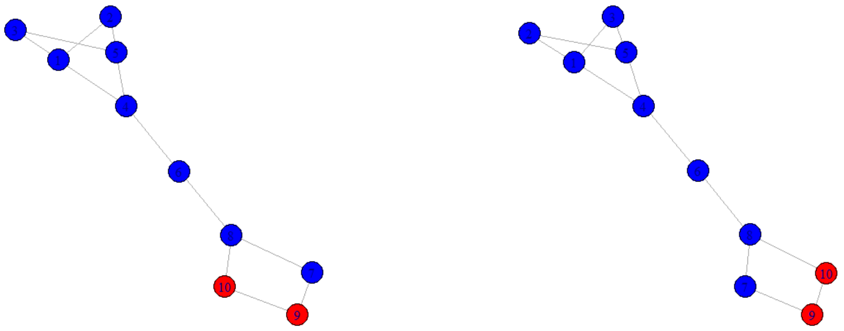
However, when a=4, b=3 but seeds = 3 and 7, no cascading occurs. Therefore, we can see that the position of seeds plays a role in determining the cascade size.

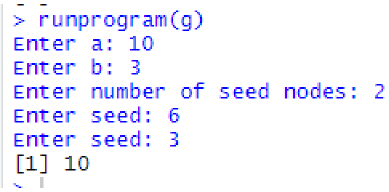


```
> runprogram(G)
Enter a: 4
Enter b: 3
Enter number of seed nodes: 2
Enter seed: 3
Enter seed: 7
[1] 2
```

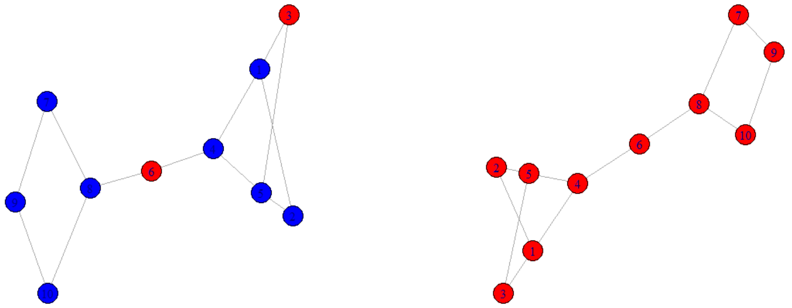


3. In the following example, for the same graph structure cascading didn't happen for low a/b vale and position of seeds in same cluster of nodes, where as high a/b value and better position of seeds gave a complete cascade.

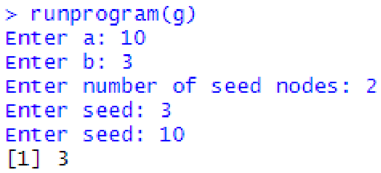




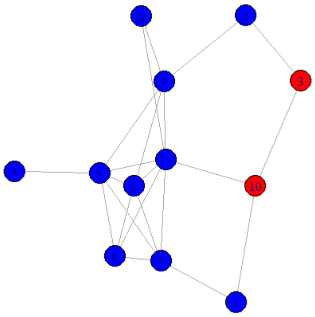
```
> runprogram(g)
Enter a: 10
Enter b: 3
Enter number of seed nodes: 2
Enter seed: 6
Enter seed: 3
[1] 10
```

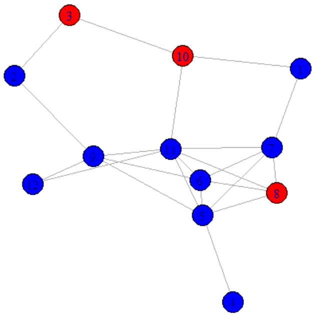


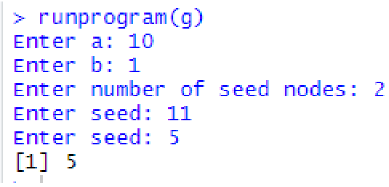
4.



```
> runprogram(g)
Enter a: 10
Enter b: 3
Enter number of seed nodes: 2
Enter seed: 3
Enter seed: 10
[1] 3
```







```
> runprogram(g)
Enter a: 10
Enter b: 1
Enter number of seed nodes: 2
Enter seed: 11
Enter seed: 5
[1] 5
```