EE232E Graphs and Network Flows
Homework #2

Sidharth Gulati – 104588717
Tushar Sudhakar Jee – 004589213
Chandan Dhal – 904588105

Introduction

Igraph is a tool used to generate graphs and networks. In this homework we used "netrw" package, which simulates the random walks of multiple walkers on a complex network, and measures the visit probability for each node of the network. We simulated random walk on random networks based on Erdos-Renyi model and Barabasi-Albert model and also calculated the average path length and the standard deviation of the path length for each walker time steps. In later part we also analyzed pagerank algorithm by random walk on different types of network and discussed more about our personalized PageRank algorithm.

Exercise #1 :

Random Walks on Erdos-Renyi model network.

We generated a random network for 1000 nodes with probability of 0.01 for drawing an edge between any pair of nodes equal to 0.01. The network is depicted as below in figure 1.

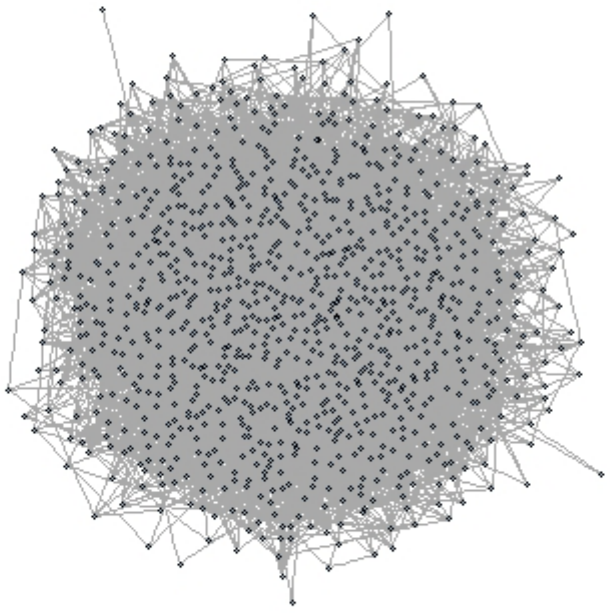

Fig 1: Erdos-Renyi model network. It can be inferred that the network are connected

The various network parameters are given below:

Parameters	**Values**
Connected	TRUE
Diameter	6

Simulating Random Walk

We defined an algorithm to perform random walk on 1000 nodes random network. The algorithm is summarized as follow.

1. 1000 random walkers start random walk from randomly selected nodes at every iterations.
2. The number of steps taken by random walker is in order of the iteration. So ith iteration will take i steps.
3. We used shortest path algorithm to compute the distance travelled by the random walker, i.e. the distance between the starting and the finish point of the random walk.
4. For each iteration, we computed the average distance data point (<s(t)>) for the iteration (t = i). Here t denotes the number of steps taken.
5. Step 1 to 4 was repeated for 1000 iterations to get 1000 distance data points.

The plot below depicts the calculated mean and standard deviation of s(t).

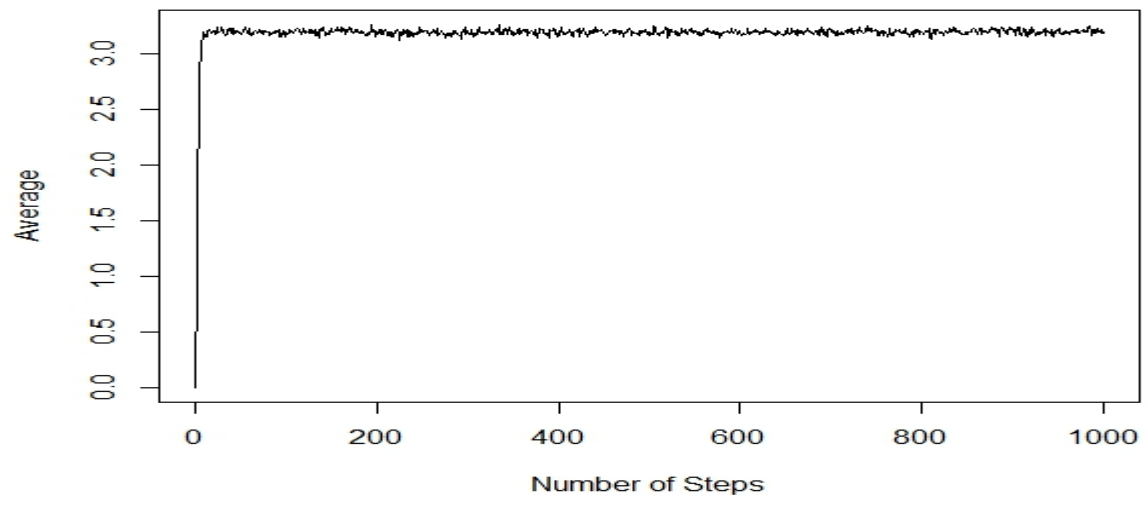

Fig 2: Average of s(t).

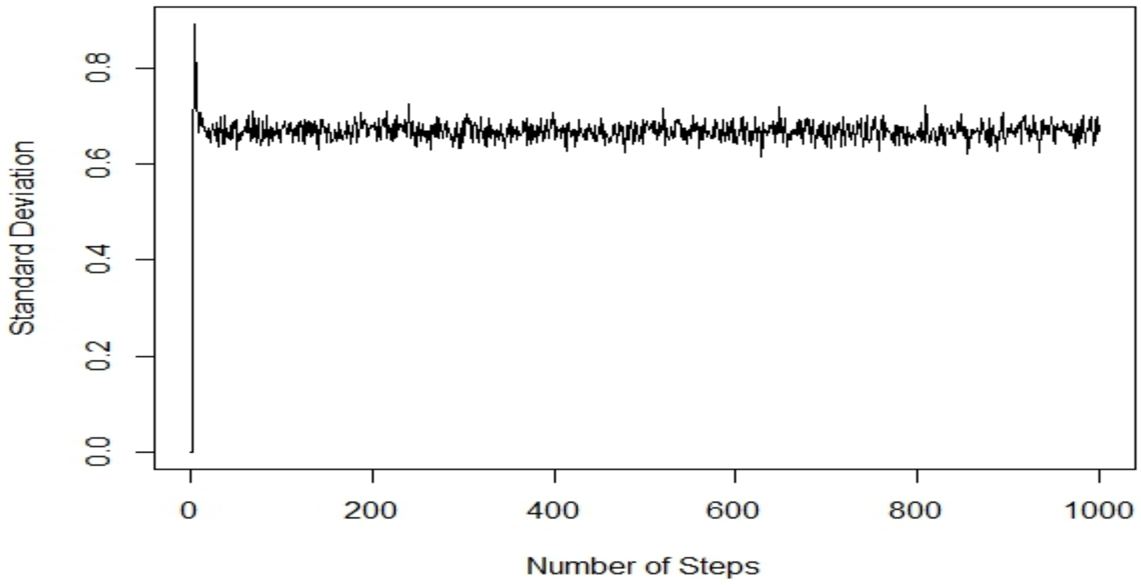

Fig 3: Standard Deviation of s(t).

The steady state of average path length can be sumarized in the given equation.

$$Average\ Path\ length\ (Erdos\ Renyi)\ =\ ln(N)/ln(Np)$$

Here N is the number of nodes in the network and p is the probability for drawing an edge between nodes. Hence for the 1000 node network with p=0.01, the average path length is 3.

From the above plots, we can infer that the average path length reaches a steady state value of 3 approximately and the standard deviation is 0.7 approximately, which is almost same to the calculated value using the above equation.

Comparison with d-dimensional network

We can observe that the random walk has a positive average distance unlike d dimensional random walker where the average distance is equal to 0. This anomaly is due to the fact that there is no notion of positive and negative distance in a random network or any network walker, unlike the d dimensional random walker. For example, in a 1-dimensional random walk if the endpoint is 1 units to the left of the start point, then the distance from the start point is assigned a value of -1, but there is no notion of negative distance in random walk.

Simulating random walk on 100 nodes network

We followed the similar algorithm to perform random walk on 100 nodes. Figure 4 depicts the 100 nodes network. The various network parameters are given below:

Parameters	Values
Connected	False
Diameter	15

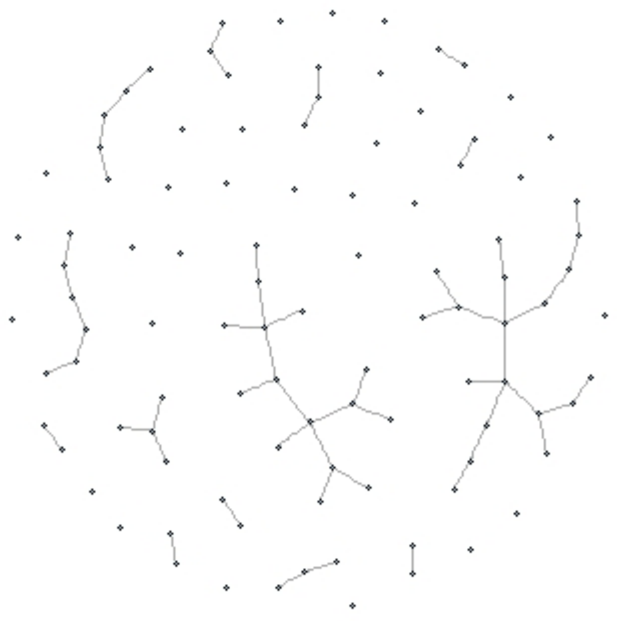

Fig. 4 Network for 100 nodes.

The plot below depicts the calculated mean and standard deviation of s(t) for 100 nodes.

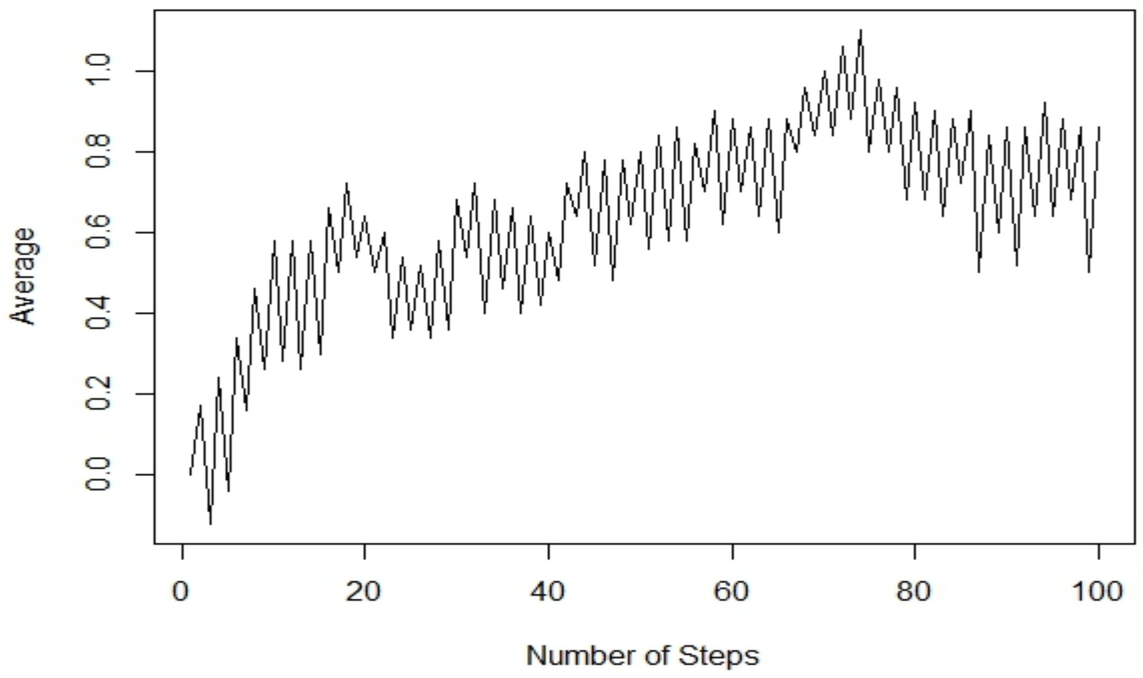

Fig 5: Average of s(t) for 100 node network.

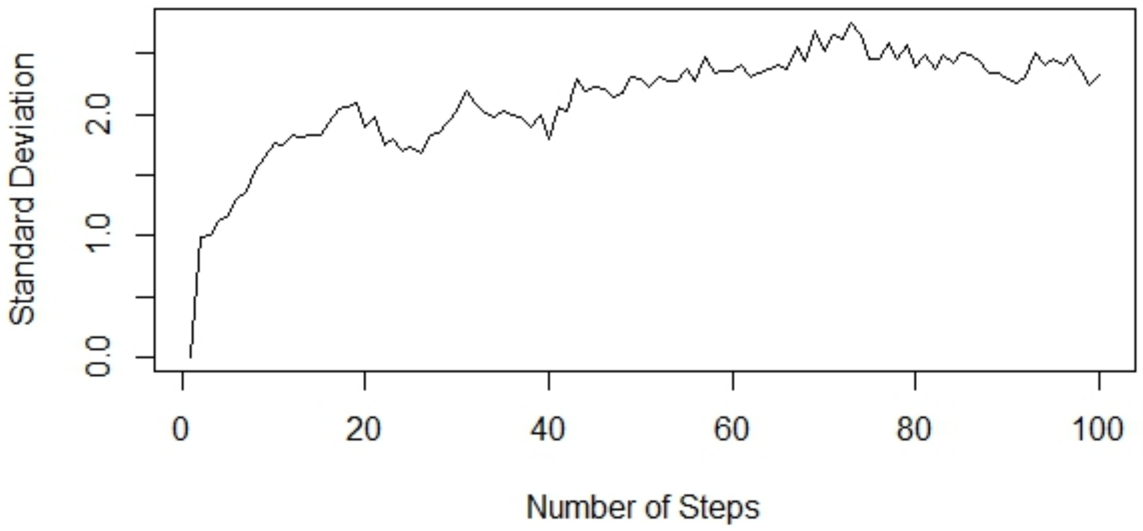

Fig 6: Standard Deviation of s(t) on 100 nodes network.

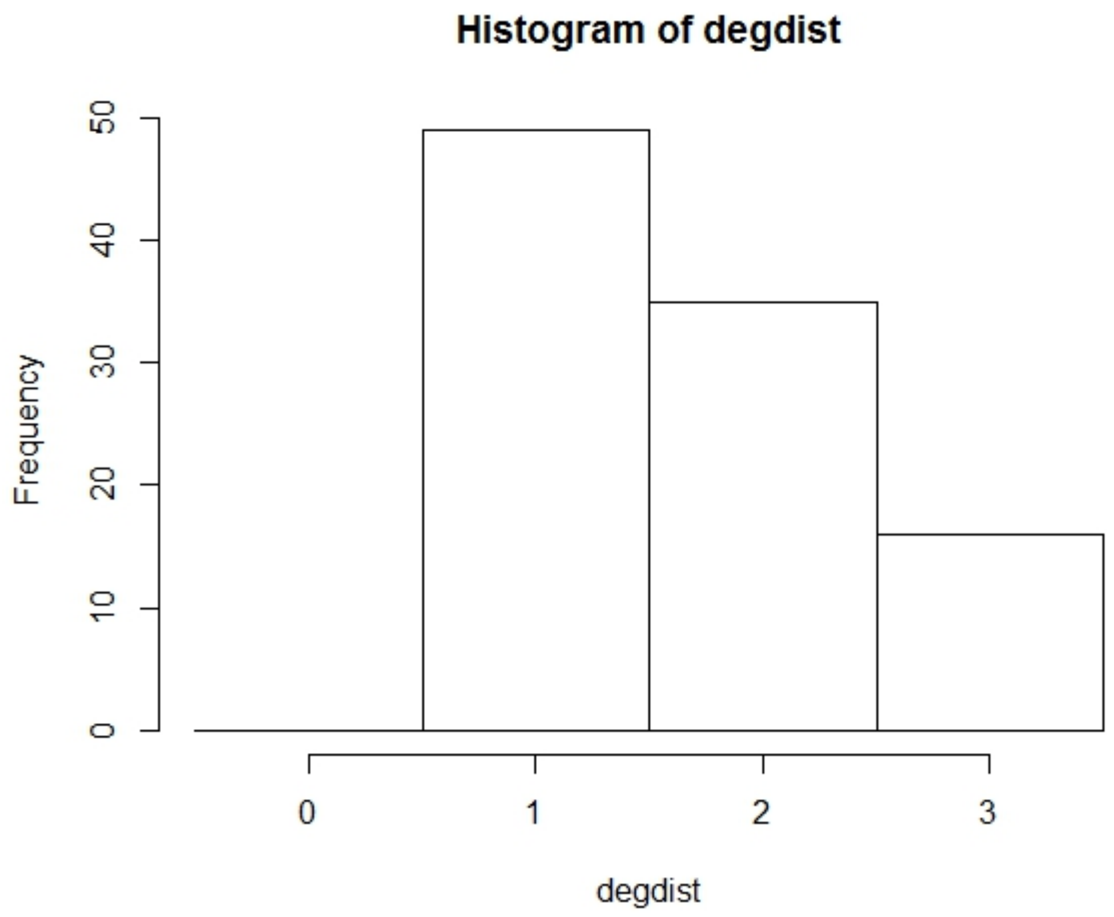

Fig 7: Histogram of degree distribution at the end of random walk

Simulating random walk on 10000 nodes network

Due to the complicacy in the huge number of iterations involved with 10,000 nodes, we modified the algorithm to perform only for 200 iterations instead of 10,000. The modified algorithm is summarized as below:-

1. 1000 random walkers start random walk from randomly selected nodes at every iterations.
2. The number of steps taken by random walker is in order of the iteration. So ith iteration will take i steps.
3. We used shortest path algorithm to compute the distance travelled by the random walker, i.e. the distance between the starting and the finish point of the random walk.
4. For each iteration, we computed the average distance data point ($<s(t)>$) for the iteration ($t = i$). Here t denotes the number of steps taken.
5. Step 1 to 4 was repeated for 1000 iterations to get 200 distance data points.

The various network parameters are given below:

Parameters	**Values**
Connected	True
Diameter	3

The plot below depicts the calculated mean and standard deviation of s(t) for 10000 nodes.

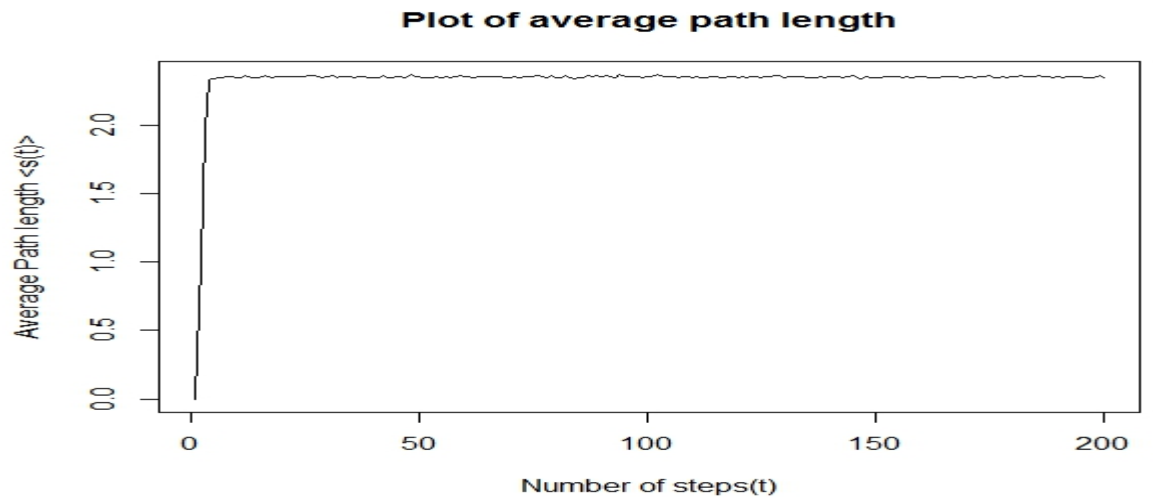

Fig 8: Average of s(t) for 10000 node network.

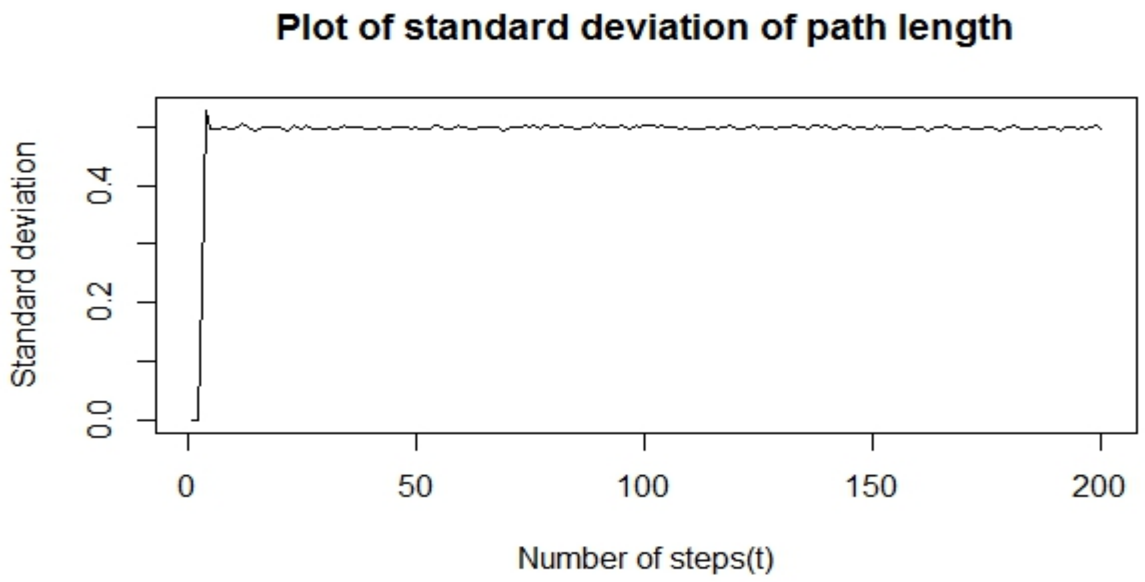

Fig 9: Standard Deviation of s(t) on 10000 nodes network.

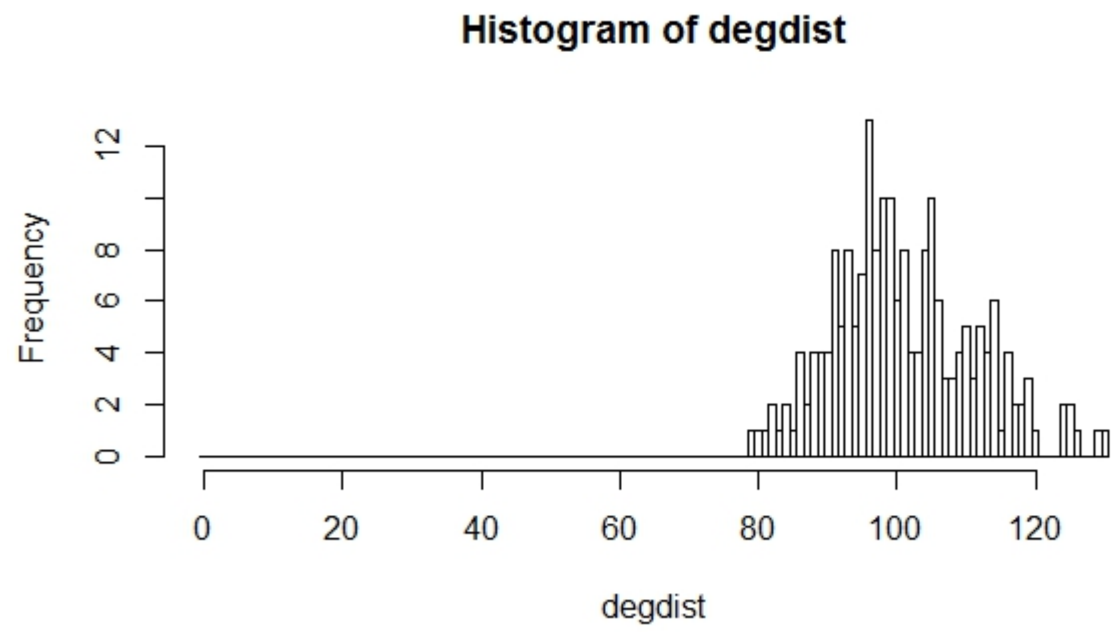

Fig 10: Histogram of degree distribution at the end of random walk

Effect of diameter of the network in calculating average and standard deviation of the distance.

In our algorithm we used shortest path algorithm to find the distance covered during a random walk. The shortest path algorithm finds the shortest path between the start and end points and returns the number of edges traversed in the shortest path as the distance. Therefore, the average distance of random walk on networks is non-zero. We also inferred that the average distance of random walk on networks is proportional to the diameter of the network. The average path length for 1000 and 10000 nodes are tabulated below.

Number of nodes	Average path length
1000	3.1
10000	2.3

Hence if the network size increases the diameter decreases and the average path length decreases. The table above is in accordance to this observation.

Degree Distribution of the network and at the end of random walk.

The figures below depict the histogram of degree distribution at the end of the random walk and the histogram of the degree distribution of the network of 1000 nodes.

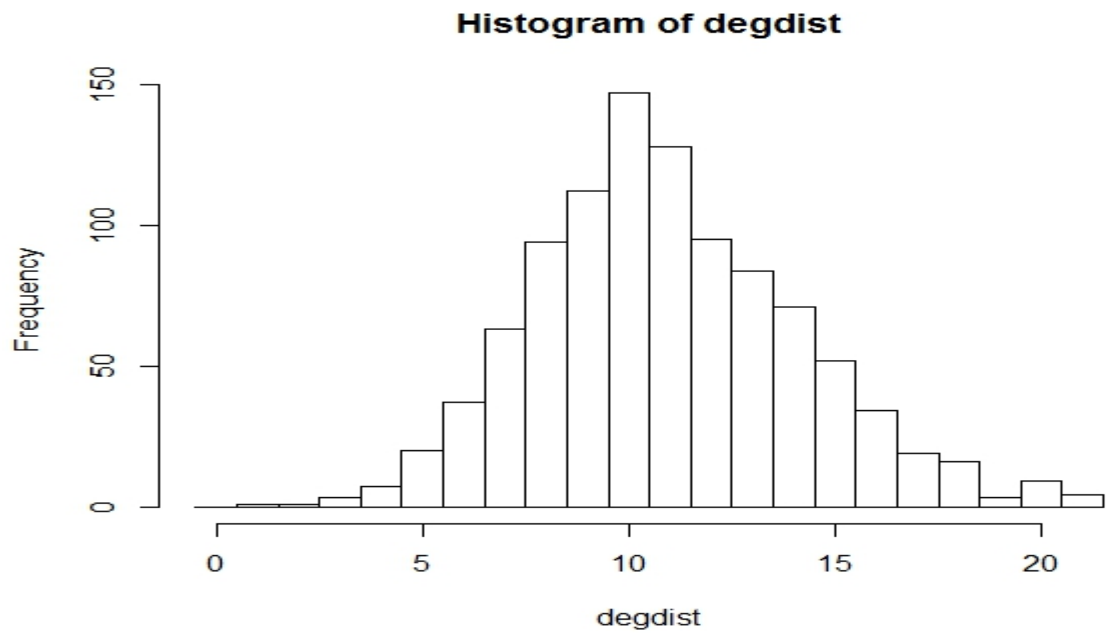

Fig 11: Histogram of degree distribution at the end of the random walk

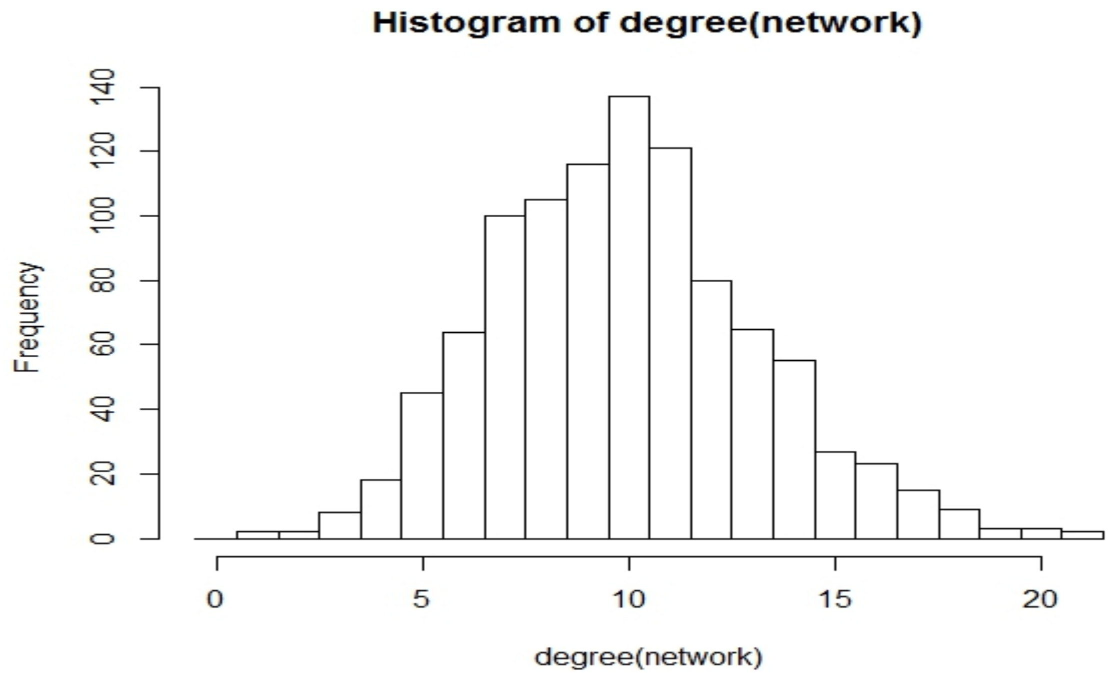

Fig 12: Histogram of degree distribution of the network

From the above plots, we can observe that the degree distribution of the nodes reached at the end of the random walk (Figure 11) closely resembles the degree distribution of the nodes in the network (Histogram of degree (Figure 12). This is due to the fact, that the random walk almost covered the entire network during the 1000 iterations for 1000 random walkers at each iteration. As a result, the walkers visited almost all the nodes in a network and hence the degree distribution of the nodes covered by the random walk matches closely with the degree distribution of the nodes in the network.

<u>**Exercise #2:**</u>
<u>***Random Walks on Fat-Tailed degree distribution model or Barabasi network.***</u>

We generated a random network for 1000 nodes with probability of 0.01 for drawing an edge between any pair of nodes equal to 0.01. The network is depicted as below in figure 13.

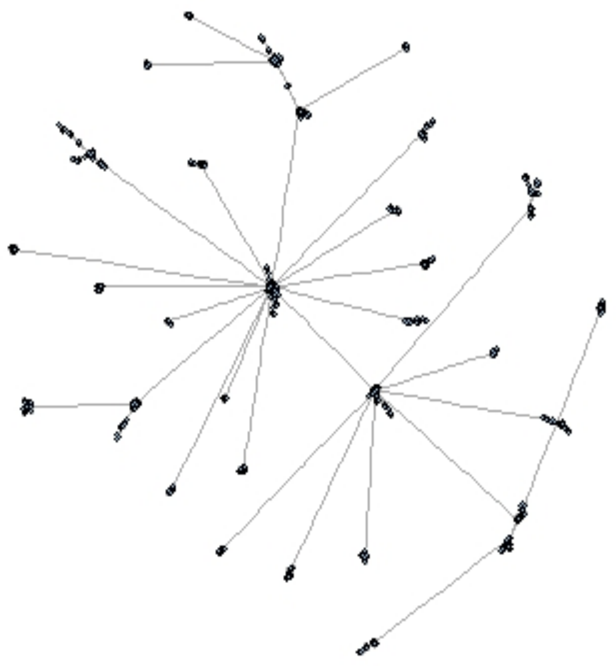

Fig 13: Barabasi model Network. It can be inferred that the network are connected

The various network parameters are given below:

Parameters	**Values**
Connected	TRUE
Diameter	15

Simulating Random Walk

We defined an algorithm to perform random walk on 1000 nodes random network. The algorithm is summarized as follow.

1. 1000 random walkers start random walk from randomly selected nodes at every iterations.
2. The number of steps taken by random walker is in order of the iteration. So ith iteration will take i steps.
3. We used shortest path algorithm to compute the distance travelled by the random walker, i.e. the distance between the starting and the finish point of the random walk.
4. For each iteration, we computed the average distance data point (<s(t)>) for the iteration (t = i). Here t denotes the number of steps taken.
5. Step 1 to 4 was repeated for 1000 iterations to get 1000 distance data points.

The plot below depicts the calculated mean and standard deviation of s(t).

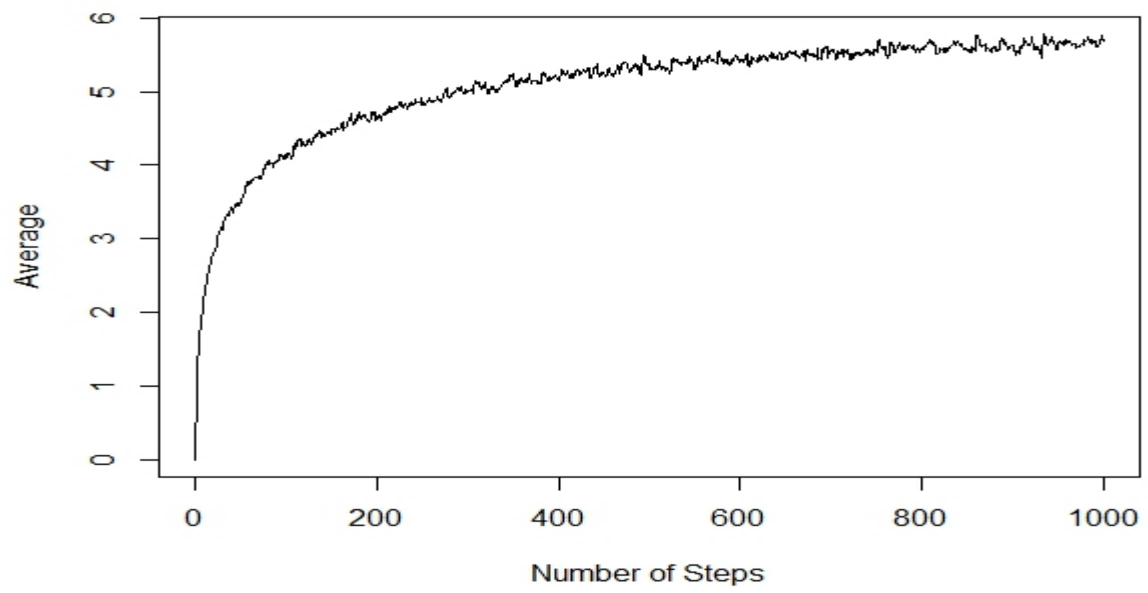

Fig 14: Average of s(t).

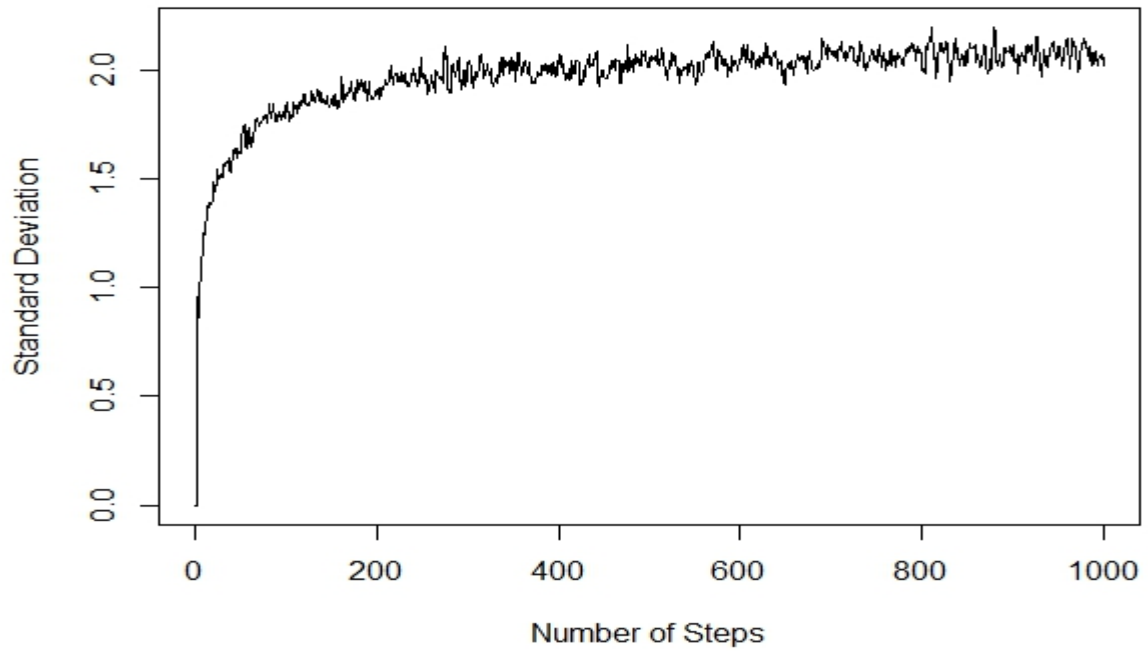

Fig 15: Standard Deviation of s(t).

The steady state of average path length can be sumarized in the given equation.

$$Average\ Path\ length\ (Barbasi)\ =\ ln(N)/ln(ln(N))$$

Here N is the number of nodes in the network. For 1000 node network the analytical value obtained is 3.7.

Comparison with dimensional network

We can observe that the random walk has a positive average distance unlike d dimensional random walker where the average distance is equal to 0. This anomaly is due to the fact that there is no notion of positive and negative distance in a random network or any network walker, unlike the d dimensional random walker. For example, in a 1-dimensional random walk if the endpoint is 1 units to the left of the start point, then the distance from the start point is assigned a value of -1, but there is no notion of negative distance in random walk.

Simulating random walk on 100 nodes network

We followed the similar algorithm to perform random walk on 100 nodes. Figure 4 depicts the 100 nodes network. The various network parameters are given below:

Parameters	**Values**
Connected	True
Diameter	9

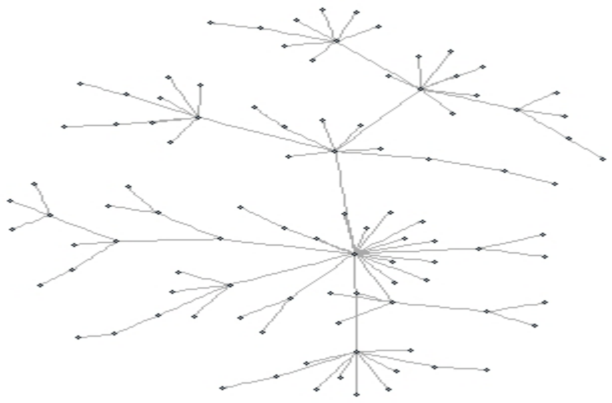

Fig. 16: Network for 100 nodes.

The plot below depicts the calculated mean and standard deviation of s(t) for 100 nodes.

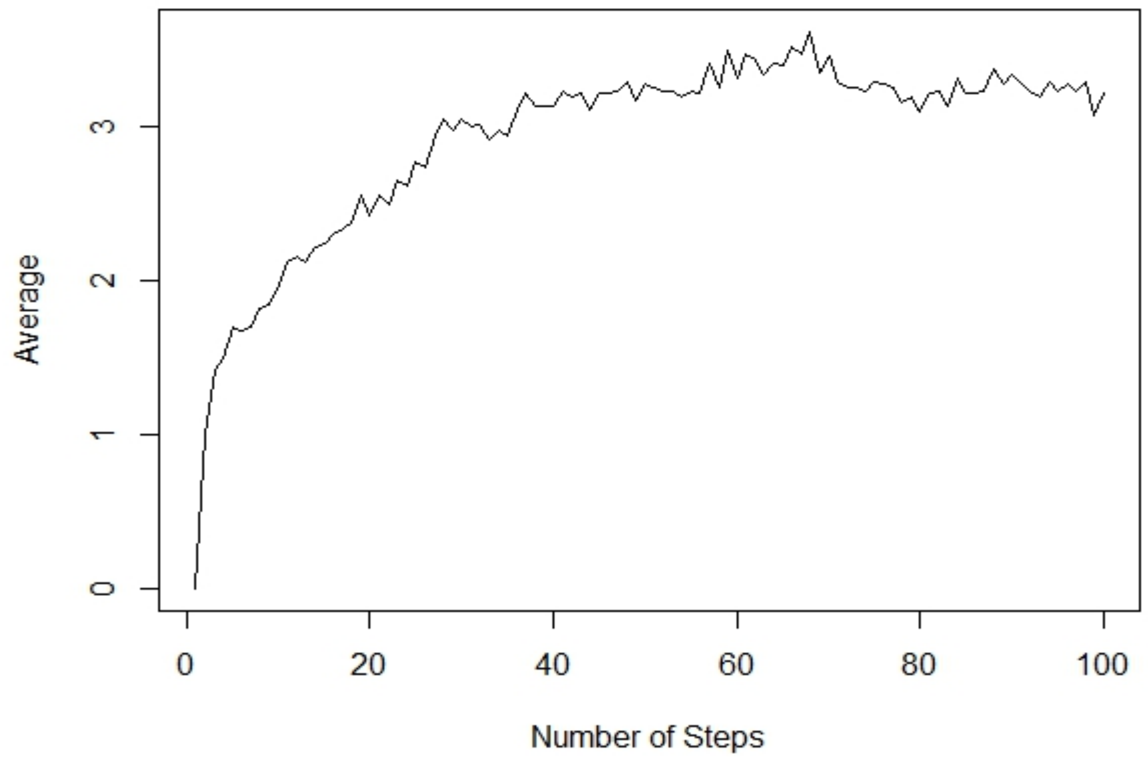

Fig 17: Average of s(t) for 100 node network.

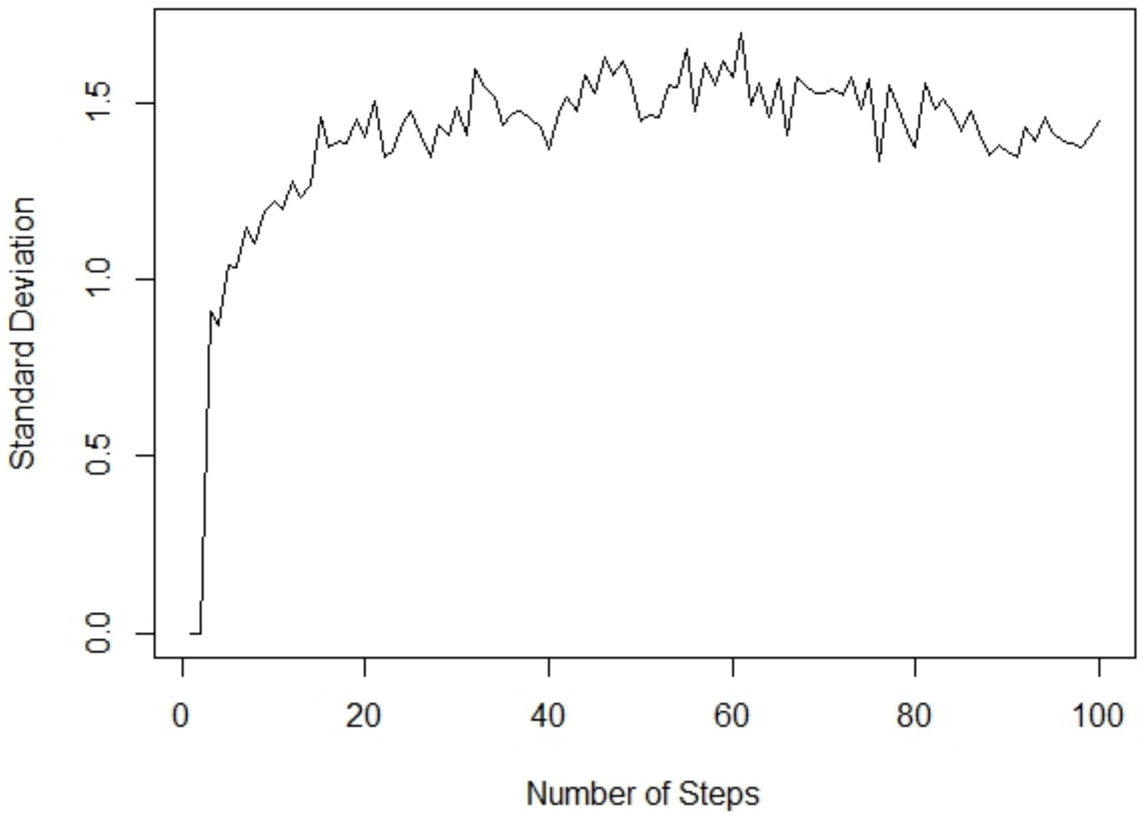

Fig 18: Standard Deviation of s(t) on 100 nodes network.

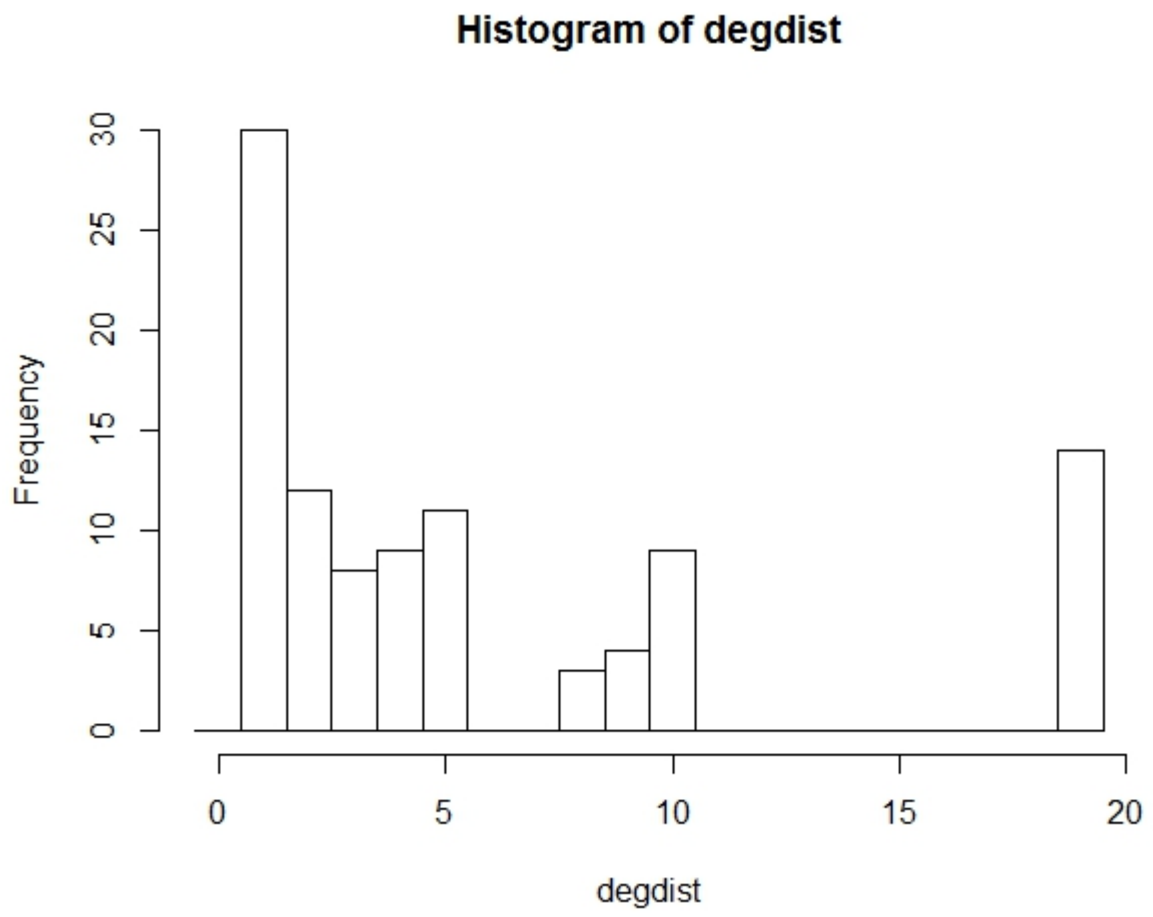

Fig 19: Histogram of degree distribution at the end of random walk

Simulating random walk on 10000 nodes network

Due to the complicacy in the huge number of iterations involved with 10,000 nodes, we modified the algorithm to perform only for 200 iterations instead of 10,000. The modified algorithm is summarized as below:-

1. 1000 random walkers start random walk from randomly selected nodes at every iterations.
2. The number of steps taken by random walker is in order of the iteration. So ith iteration will take i steps.
3. We used shortest path algorithm to compute the distance travelled by the random walker, i.e. the distance between the starting and the finish point of the random walk.
4. For each iteration, we computed the average distance data point (<s(t)>) for the iteration (t = i). Here t denotes the number of steps taken.
5. Step 1 to 4 was repeated for 1000 iterations to get 200 distance data points.

The various network parameters are given below:

Parameters	Values
Connected	True
Diameter	2

The plot below depicts the calculated mean and standard deviation of s(t) for 10000 nodes.

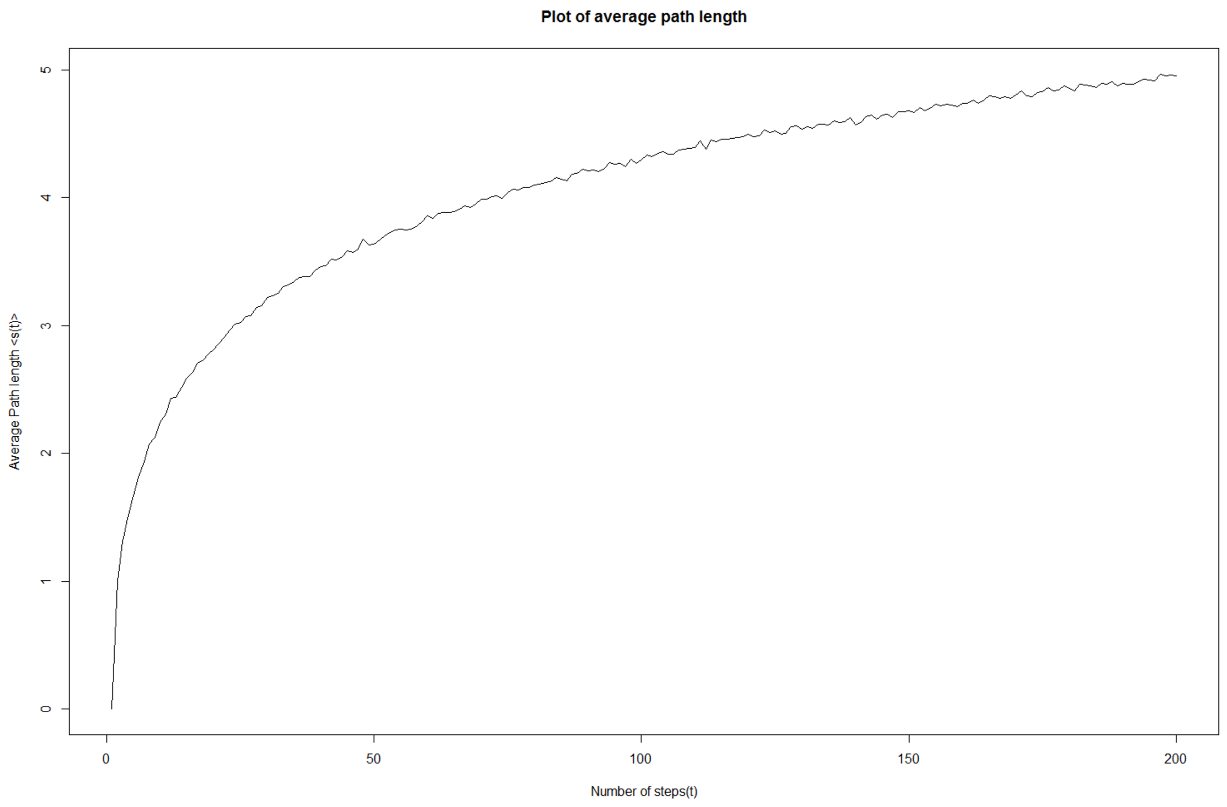

Fig 20: Average of s(t) for 10000 node network.

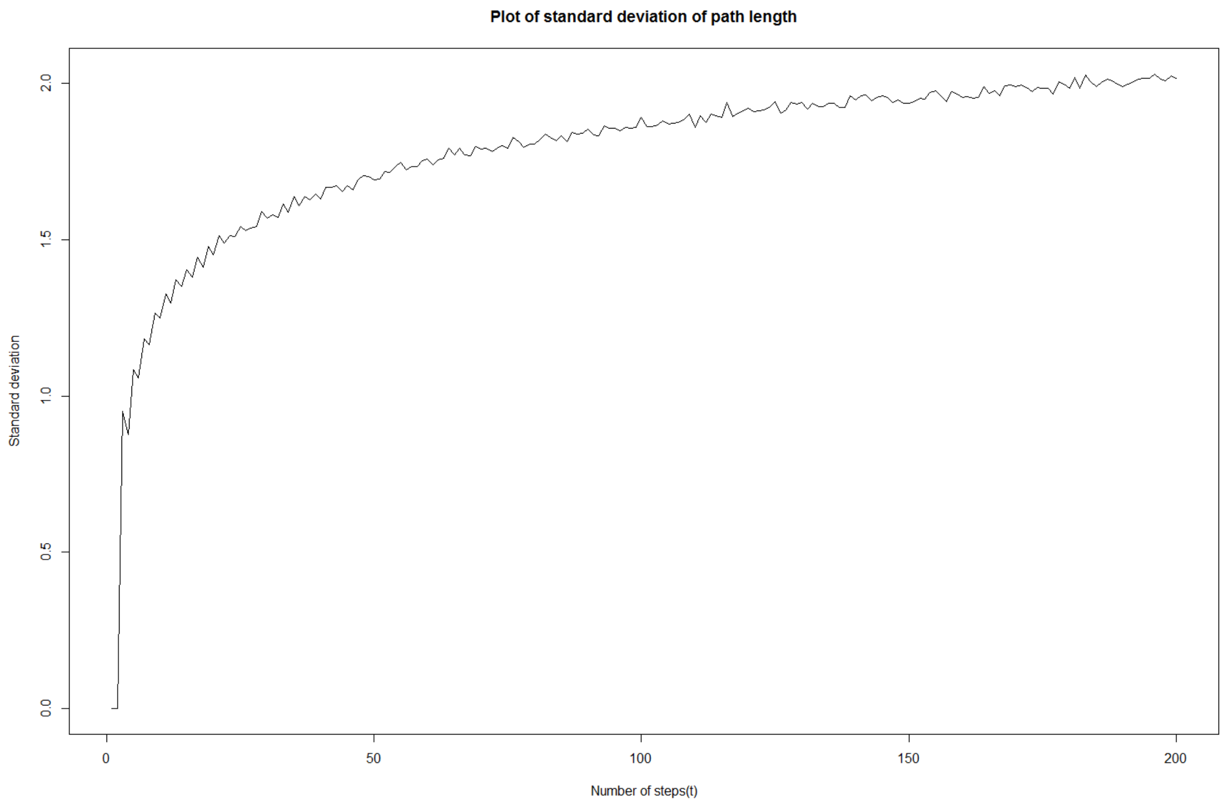

Fig 21: Standard Deviation of s(t) on 10000 nodes network.

Effect of diameter of the network in calculating average and standard deviation of the distance.

In our algorithm we used shortest path algorithm to find the distance covered during a random walk. The shortest path algorithm finds the shortest path between the start and end points and returns the number of edges traversed in the shortest path as the distance. Therefore, the average distance of random walk on networks is non-zero. We also inferred that that the average path length increases approximately logarithmically with network size. This behavior is also reflected in the simulation results tabulated below

Number of nodes	Average path length
100	3.3
1000	3.9
10000	4.6

Hence as the network size increases the diameter increases and since average path length is of the order of the diameter so that the average path length also increase.

Degree Distribution at the end of random walk.

The figures below depict the histogram of degree distribution at the end of the random walk and the histogram of the degree distribution of the network of 1000 nodes.

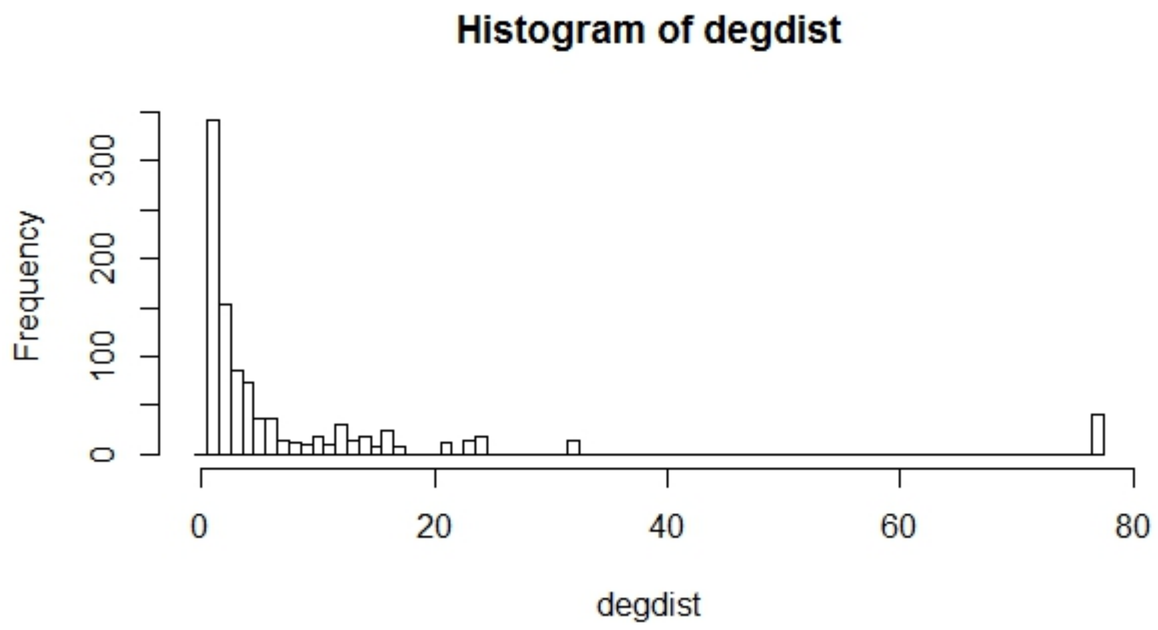

Fig 22: Histogram of degree distribution at the end of the random walk

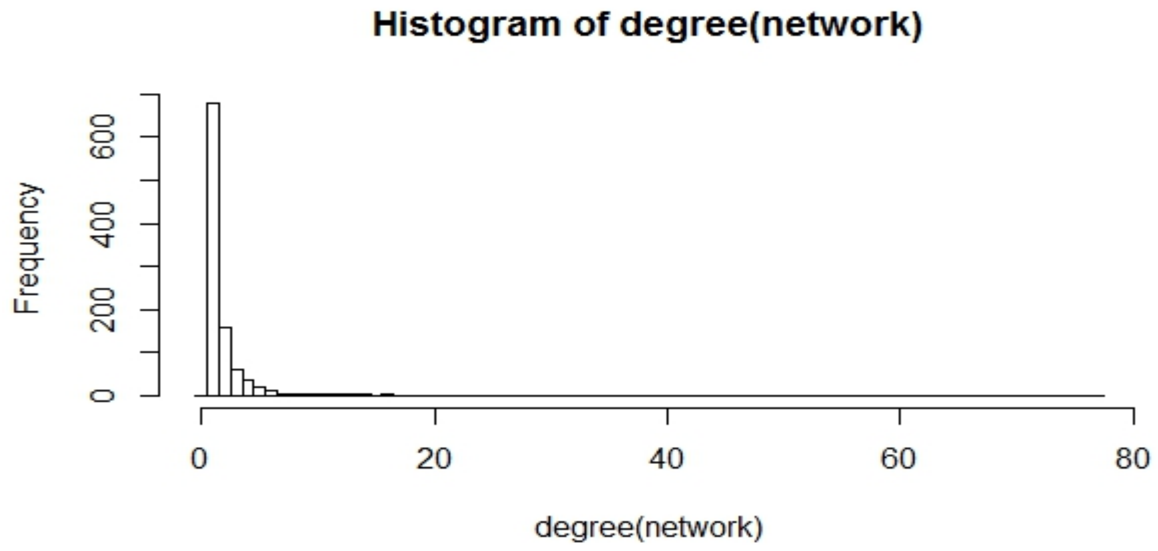

Fig 23: Histogram of degree distribution of the network

From the above plots, we can observe that the degree distribution of the nodes reached at the end of the random walk (Figure 22) closely resembles the degree distribution of the nodes in the network (Histogram of degree (Figure 23). This is due to the fact, that the random walk almost covered the entire network during the 1000 iterations for 1000 random walkers at each iteration. As a result, the walkers visited almost all the nodes in a network and hence the degree distribution of the nodes covered by the random walk matches closely with the degree distribution of the nodes in the network.

Exercise #3:

Page rank and random walk:

In the last decade, PageRank has emerged as a very powerful measure of relative importance of nodes in a network.PageRank is used in a variety of domains like Data mining, Web algorithms, and Distributed computing for determining important or similar nodes, load balancing, search,identifying connectivity structures and ranking importance of webpages on the Web.

Below are the algorithms for directed or undirected graph and undirected graph.The former takes O(log n/ε) rounds with high probability on any graph (directed or undirected), where n is the network size and ε is the reset probability.The latter is a faster algorithm taking O(√log n/ε) rounds in undirected graphs.

The BASIC-PAGERANK-ALGORITHM(O(log n/ε)) for directed or undirected network is as:

Algorithm 1 BASIC-PAGERANK-ALGORITHM

Input (for every node): Number of nodes n and reset probability ϵ.

Output: Approximate PageRank of each node.

```
1:  Each node $v$ maintains a count variable "$couponCount_v$" corresponding to number of random
    walk coupons held by $v$. Initially, $couponCount_v = K$ for starting $K$ random walks.
2:  Each node $v$ also maintains a counter $\zeta_v$ for counting the number visits of random walks to it.
    Set $\zeta_v = K$.
3:  for round $i = 1, 2, \ldots, B \log n/\epsilon$ do   //[for sufficiently large constant $B$]
4:     Each node $v$ holding at least one alive coupon (i.e., $couponCount_v \neq 0$) does the following
    in parallel:
5:     For every neighbor $u$ of $v$, set $T_v^u = 0$        // [$T_v^u$ is the number of random walks moving
    from $v$ to $u$ in round $i$]
6:     for $j = 1, 2, \ldots, couponCount_v$ do
7:        With probability $1 - \epsilon$, pick a uniformly random outgoing neighbor $u$
8:        $T_v^u := T_v^u + 1$
9:     end for
10:    Send the coupon counter number $T_v^u$ to the respective outgoing neighbors $u$.
11:    Each node $u$ computes: $\zeta_u = \zeta_u + \sum_{v \in N(u)} T_v^u$.       //[the quantity $\sum_{v \in N(u)} T_v^u$ is the
    total number of visits of random walks to $u$ in $i$-th round (from its neighbors)]
12:    Each node $u$ update the count variable $couponCount_u = \sum_{v \in N(u)} T_v^u$
13: end for
14: Each node $v$ outputs its PageRank as $\frac{\zeta_v \epsilon}{cn \log n}$.
```

PageRank is a technique to estimate the importance of a node in a directed or undirected graph based on the stationary distribution of a particular random walk in the graph. This random walk models a memoryless random surfer browsing around the web. Once at a node in the directed graph, the random surfer has two choices:

1. Move to a new node from the set of out-edges, or
2. Do something else (like close the browser, enter a search query, navigate to a bookmark etc)

The probability of the first action is commonly known as the *damping parameter* in PageRank. The second action is called *teleporting*. Usually, this choice is modeled by the surfer picking a node at random according to a chosen distribution.

Relation between visit probability and node degree (undirected,d=1):

In the random walk simulation on a 1000 node network (undirected), we measured the probability of the walker visiting each node. We also computed the degree of the nodes. In order to get a sense of the relationship between the visit probability and the node degree, we plotted them. The plot is given below

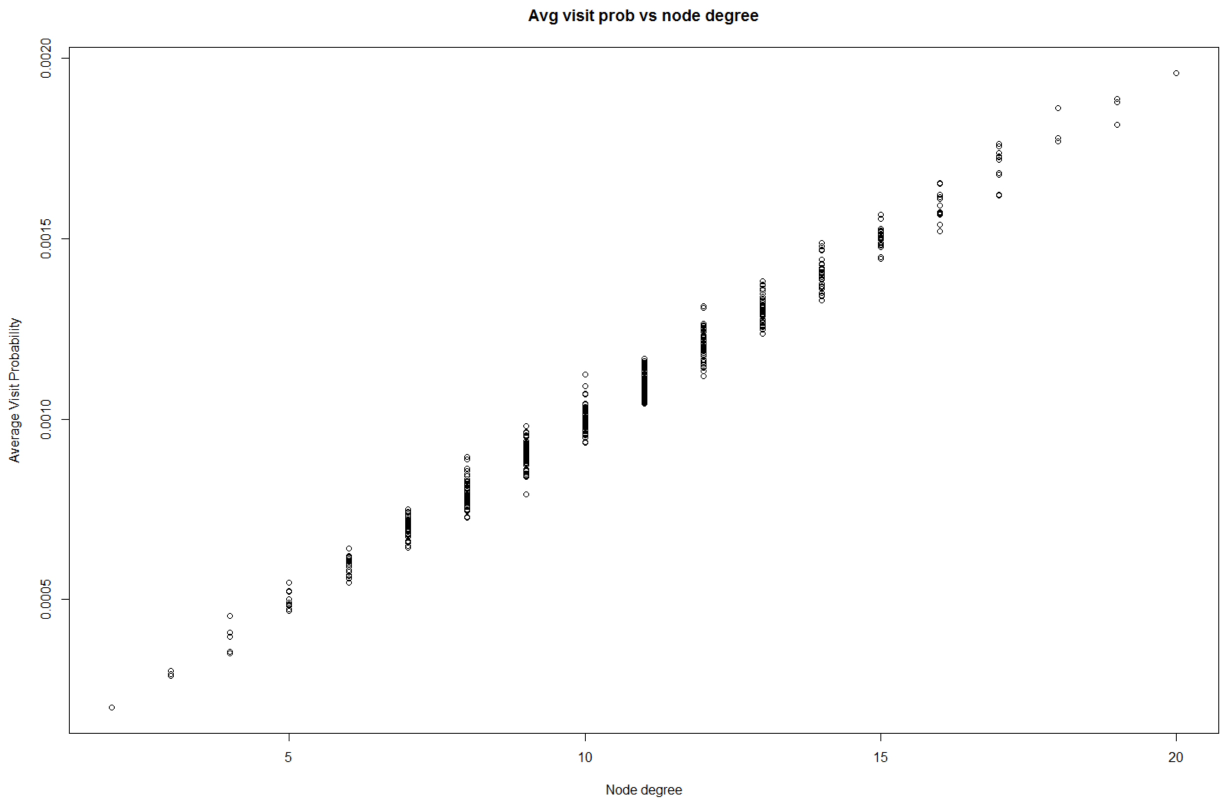

Fig 24 Visit probability and Node degree (undirected)

From the above plot it can be seen that the visit probability is linearly (almost) related to the node degree. The higher the degree of the node the larger the probability of the walker visiting that node.

Relation between visit probability and in-node degree (directed,d=1):

In the random walk simulation on a 1000 node network (directed), we measured the probability of the walker visiting each node. We also computed the in-degree of the nodes. In order to get a sense of the relationship between the visit probability and the in-node degree, we plotted them. The plot is given below

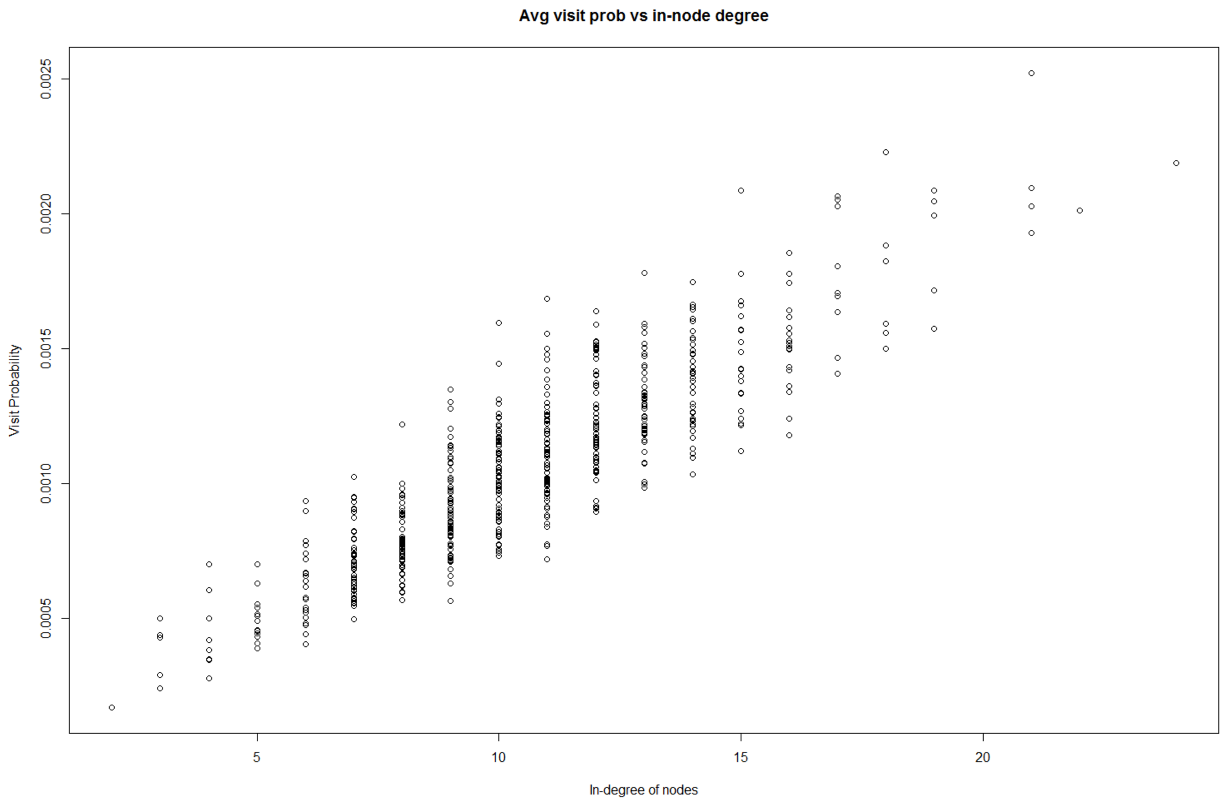

Fig 25 Visit probability and In-Node degree(directed)

From the above plot it can be seen that the visit probability is linearly (almost) related to the in-node degree. The higher the in-degree of the node the larger the probability of the walker visiting that node.

Relation between visit probability and in-node degree (undirected,d=0.85,tel=NULL):

In the random walk simulation on a 1000 node network (undirected), with damping and teleportation = NULL (default), we measured the probability of the walker visiting each node. We also computed the degree of the nodes. In order to get a sense of the relationship between the visit probability and the node degree, we plotted them.

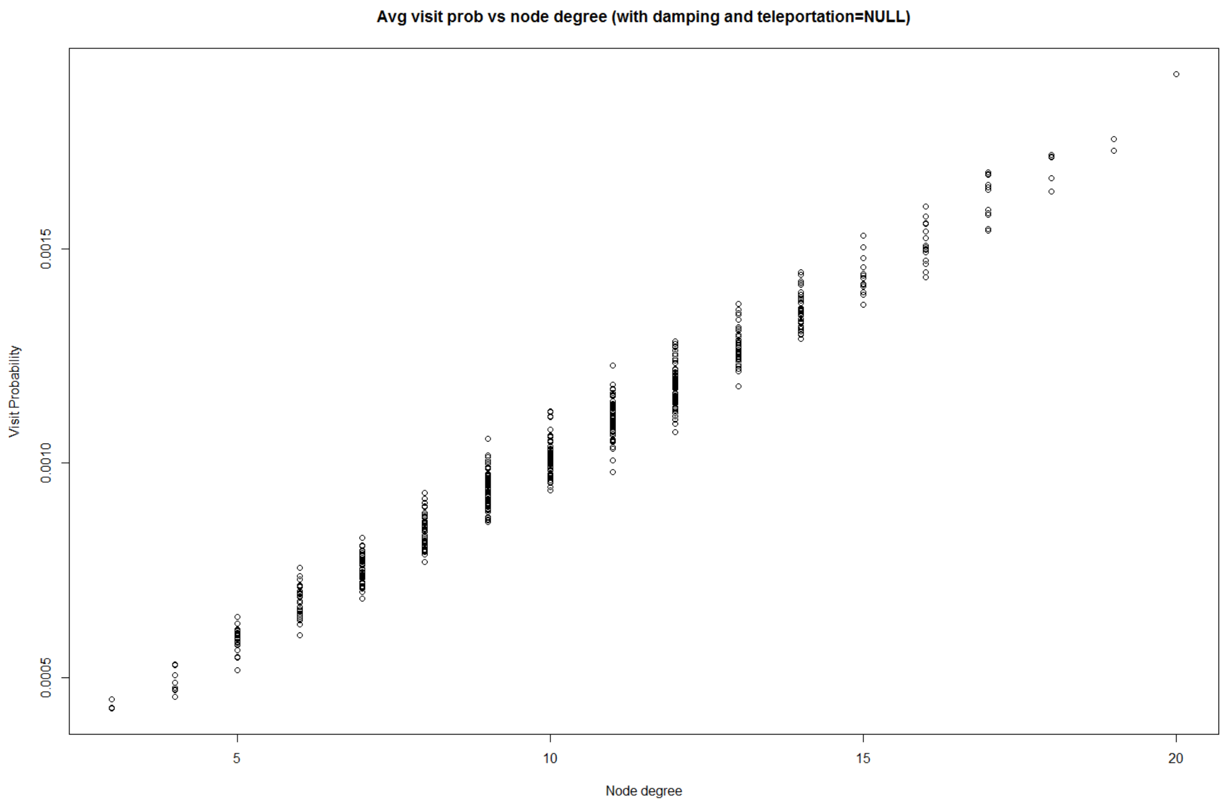

Fig 26 Visit probability and Node degree(No Teleportation)

From the above plot it can be seen that the visit probability is linearly (almost) related to the node degree. The higher the degree of the node the larger the probability of the walker visiting that node. Also a key observation is that the plot with and without damping does not change if the teleportation vector is null or all of equal weights. We demonstrate an interesting phenomenon (to be used in customized pagerank) in the next section.

Relation between visit probability and in-node degree (undirected,d=0.85,tel=custom):

In the random walk simulation on a 1000 node network (undirected), with damping and teleportation = customized (the vector has 1's in position 5 and 6 and the rest are 0's), we measured the probability of the walker visiting each node. We also computed the degree of the nodes. In order to get a sense of the relationship between the visit probability and the node degree, we plotted them. The plot is shown below

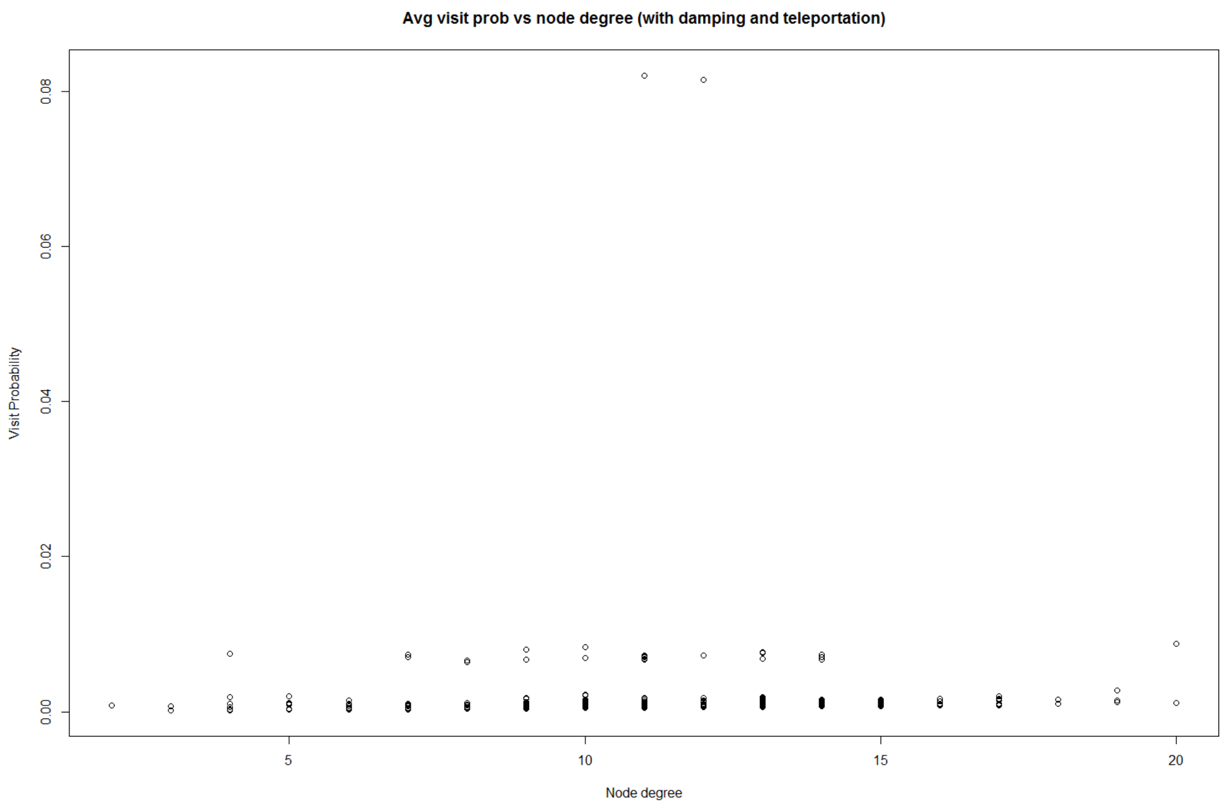

Fig 27 Visit probability and Node degree(customize teleportation)

From the above plots, we can see that all the nodes have similar visit probabilities except two nodes. These are the nodes which were set to 1's in the teleportation probability vector. Every Time a random walker decides to stop and restart his random walk (happens with probability 0.15), he chooses from those two nodes to restart his random walk. Therefore, the probability of visiting those two nodes are much higher compared to the other nodes.

Exercise #4:
Personalized PageRank

In previous parts, we showed how often a nodes are visited in a random walk depends on the in degree of that particular node. This idea is in essence the underlying mechanism of the PageRank algorithm. This algorithm is used by search engines such as Google to rank how relevant or reliable web pages are in terms of how many other websites link to that particular page. If we think about the internet as one big connected directed network, where each node is a website or webpage, then this measure is exactly represented by the in degree of a particular node. The underlying idea is that people would only link to other web pages or nodes if those web pages are reliable and useful. However a purely random walker would not model the propensity of a user to stop following hyper links and jump to an unlinked page. To account for this fact, a damping factor 'alpha' is included to model the probability of a user continuing to follow chain of links from a page. In the regular PageRank algorithm, if a crawler does decide to jump to a random unconnected node, then the probability of jumping to any node in the graph is the same. The equation that calculates the Page rank is as follows:

$$P_i = (1-\alpha)\sum(A_{ji}P_j/k_j^o) + (\alpha/N)\sum_{i=1}^{N}(P_i)$$

For the first part of this question we generate a random directed graph with 1000 nodes using the Erdos-Renyi model. Then a random walk with 1000 walkers for 1000 steps is performed. The damping factor is set to 0.85 and the teleportation probability for all nodes is the same. At the end point we calculate the average probability that a node was visited in a walk. This is plotted against the degree of each node. The plot is shown below:

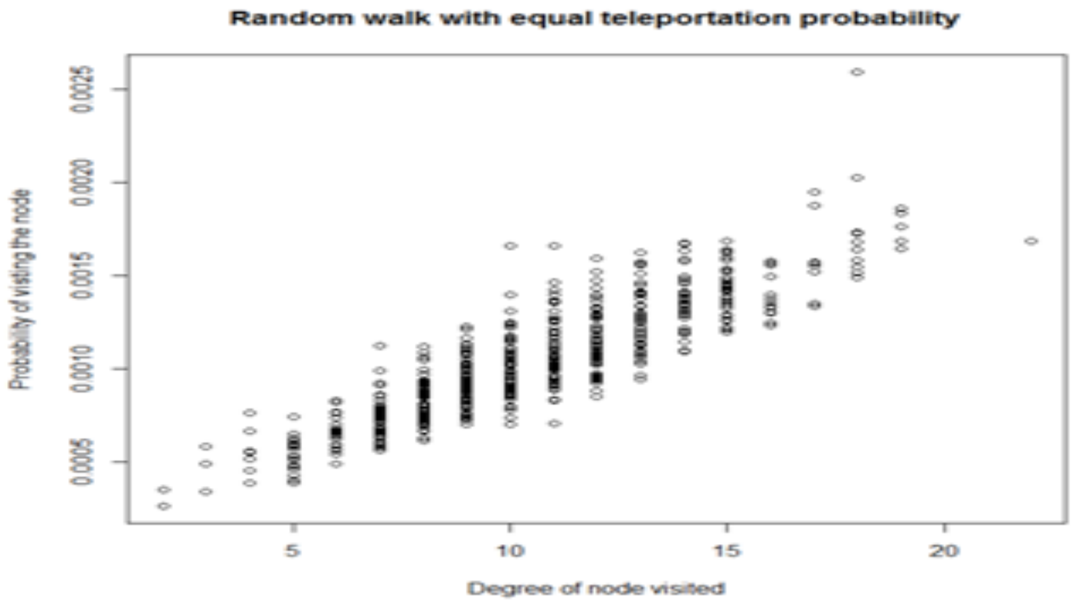

Fig 28 Probability of visiting nodes vs Degree

We can clearly see that as the in degree of the nodes increases, probability of visiting that particular node increases. Note there are several data points for some degree values as the graph has many nodes with the same degree. In general we see that nodes with higher in degrees tend to have higher probabilities of being visited.

Next we use the page.rank function in R to calculate the page rank for each node. The damping factor was set to 0.85 again. The page rank for each node is plotted against the degree of the node. The plot is as follows:

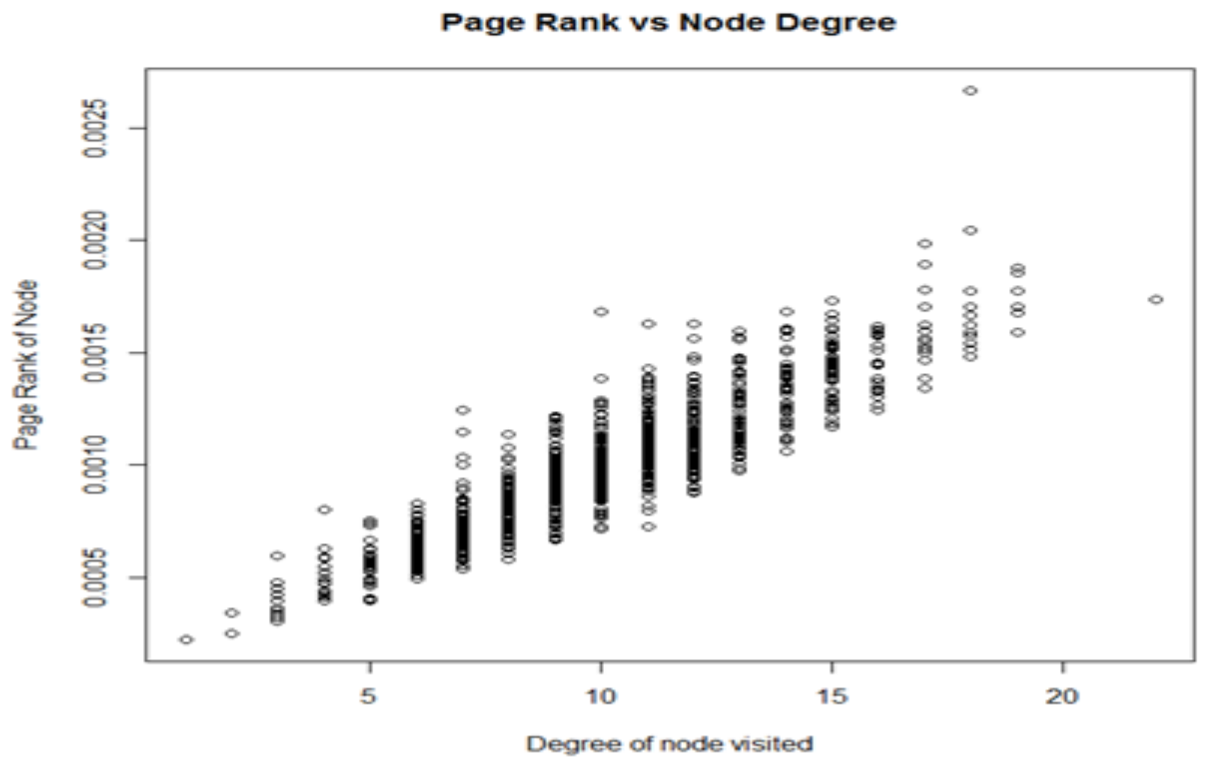

Fig 29 Rank vs Degree

We can see that a similar trend to what was observed in the first graph is also observed here. The trends in both graphs are extremely similar, with nodes of higher degrees tending to higher values of page rank. Hence we see that page rank is basically is modeled by the probably of a random walker visiting that node and is proportional to the in degree of the node.

Personalized PageRank

In this part, we study the teleportation probability in more detail. The normal page rank models the teleportation behavior such that when the user decides to randomly jump to an unconnected page, each page has an equal probability of being jumped to. This is not the case in real life as many users have their own personal likes and dislikes and jump to certain pages with a certain preference. To show how these preferences affect the rank of a node, we use the page rank generated for the same graph in part 4a as a preference vector. This preference is then used as the teleportation probability of nodes in and a random walk is performed. While the damping factor is kept the same (0.85) note that the probability that the walker jumps to some node **i** depends on that node's page rank score. Once again the probability of visiting the node is plotted against the degree of the nodes. The plot is as follows:

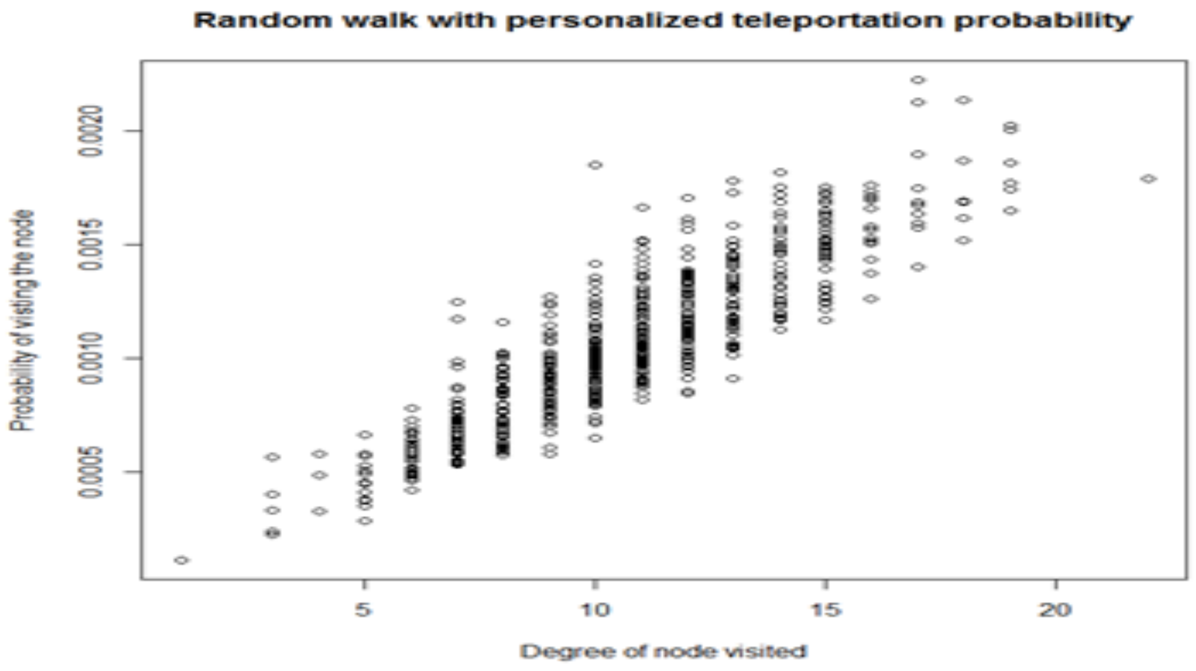

Fig 30

Here we can clearly see a shift in the visit probability as nodes with higher in degrees have a comparatively larger visit probability than previously observed. This is due to the fact that it is much more likely that when a walker teleports, it will appear at a node with high in degree increasing its visit probability and page score. To further highlight how much of an effect personal preference can have, let us try the edge case where the user teleports to one node in particular 15% of the time. In this case the teleportation vector will have zeroes for all nodes except for a random node of the walker's liking. The plot is shown below:

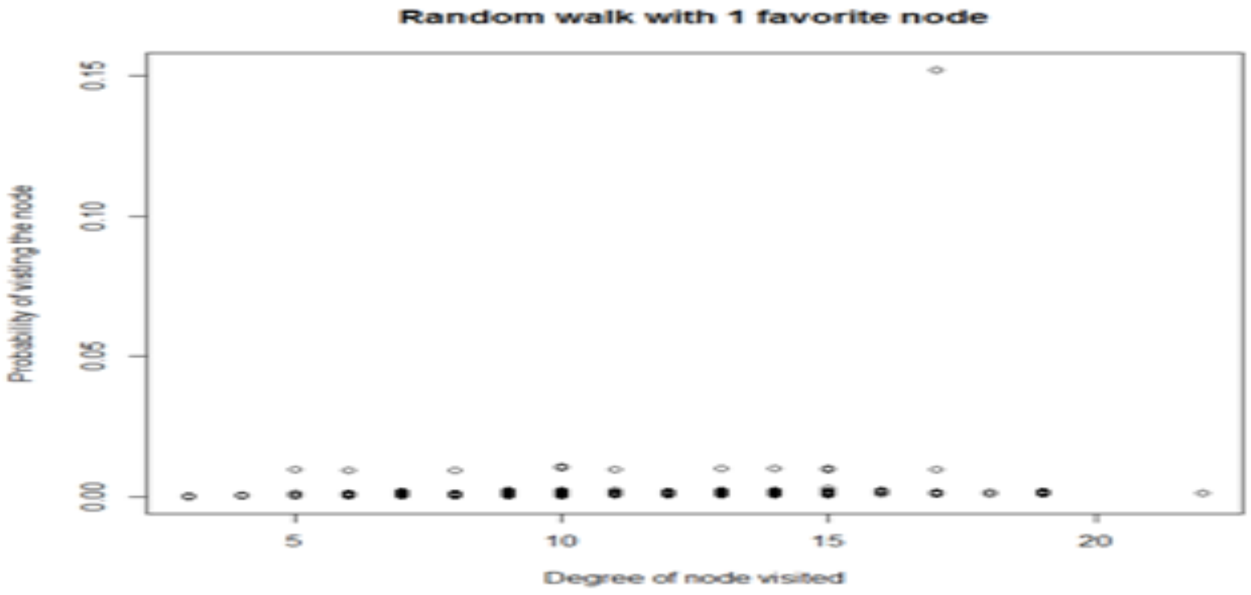

Fig 31

Here we can see that the favorite node has significantly higher probability of visit, which is expected. We can see the effect of node degree is still present if we disregard the favorite node.

To modify the previously described PageEank equation, we only need a vector of weights that we use for the personalized teleportation probability instead of using 1/N. The equation is then as follows:

$$P_i = (1-\alpha)\sum(A_{ji}P_j/k_j^o) + (\alpha)\sum_{i=1}^{N}(\sigma_i P_i)$$

$$where \sum_{i=1}^{N}(\sigma_i) = 1$$

The the variable sigma's are the weights representing how likely a user will teleport to that particular page. The above set of equations can be represented as a matrix equation which may then be solved iteratively to get the PageRank values. In practice these values may be found by using the random walking crawlers that we have seen in this report.

Conclusion

In this report, we simulated the random walk algorithm for 100, 1000 and 10000 nodes for the *Erdos-renyi model and Barabasi-Albert model* networks. For the simulations we plotted the average path length and standard deviation of the path length. We compared the analytical average path length and the simulated average path length and investigated their relation with the diameter of the network. The average path length was observed to be of the order of the diameter of the network.

For Erdos-Renyi model the diameter decreased with the increase in the size of the network (keeping p fixed) resulting in reduction in average path length. For Barabasi-Albert model the diameter increased with the increase in the size of the network, and hence the average path length increased. Further it can be concluded that the convergence to the steady state (for average path length) for Barabasi-Albert model was much slower than that of Erdos-Renyi model.

It was seen that implementations of the PageRank algorithm produced similar results to the random crawlers. Further we observed the effect of personalized weights for teleportation probability and how it closely modeled people with different preferences to crawl the web differently and produce different PageRanks.

Library References

[1] Luis Rodero-Merino, Antonio Fernández Anta, Luis López, Vicent Cholvi, *Performance of random walks in one-hop replication networks*. Computer Networks, Elsevier, 2008.

[2] Agata Fronczak, Piotr Fronczak, and Janusz A. Hołyst, *Average path length in random networks*. Physical Review, 2004.

[3] Tracking the Random Surfer: Empirically Measured Teleportation Parameters in PageRank
David F. Gleich,Paul G. Constantine,Abraham D. Flaxman and Asela Gunawardana

[4] Fast Distributed PageRank Computation
Atish Das Sarma ,Anisur Rahaman Molla ,Gopal Pandurangan and Eli Upfal.