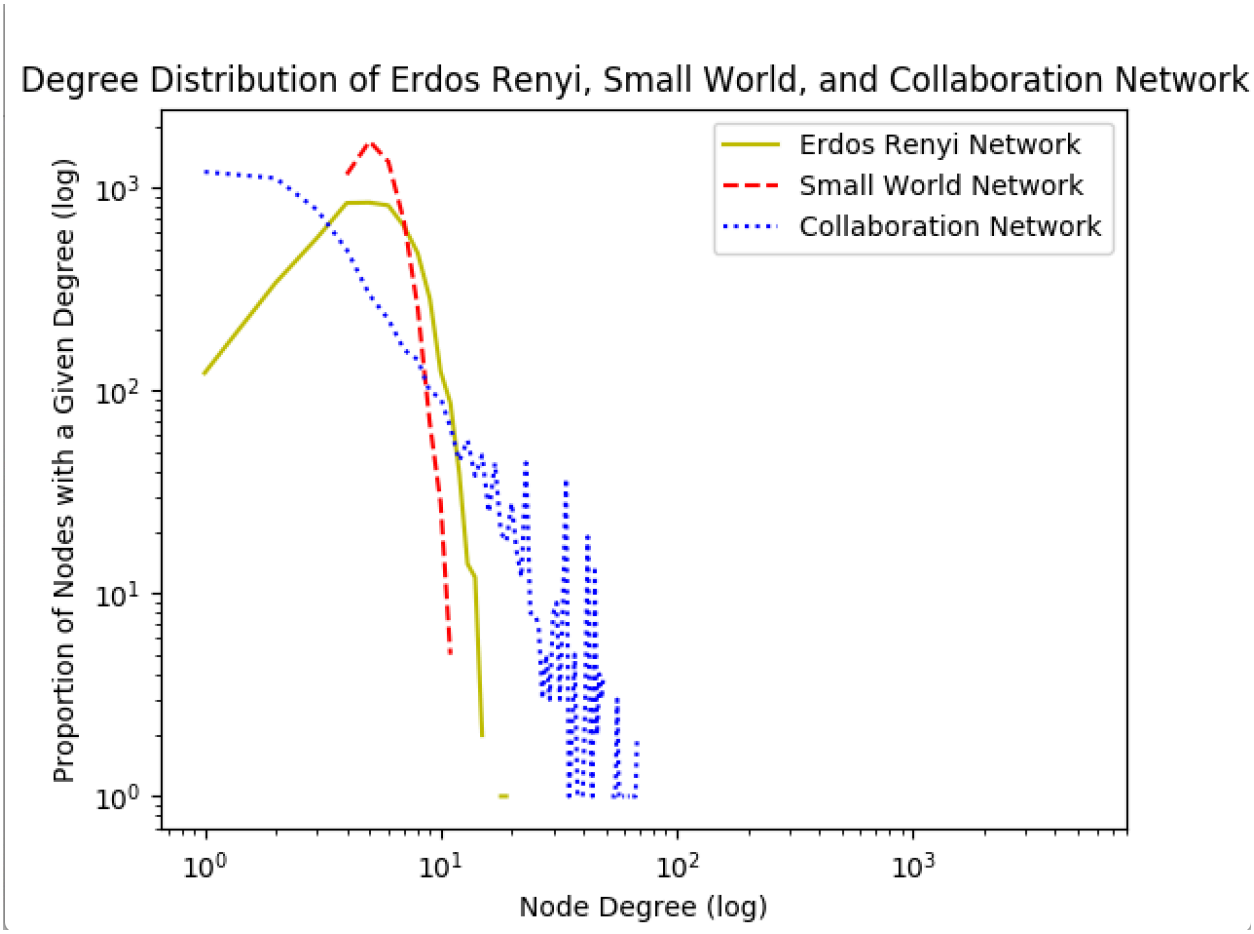
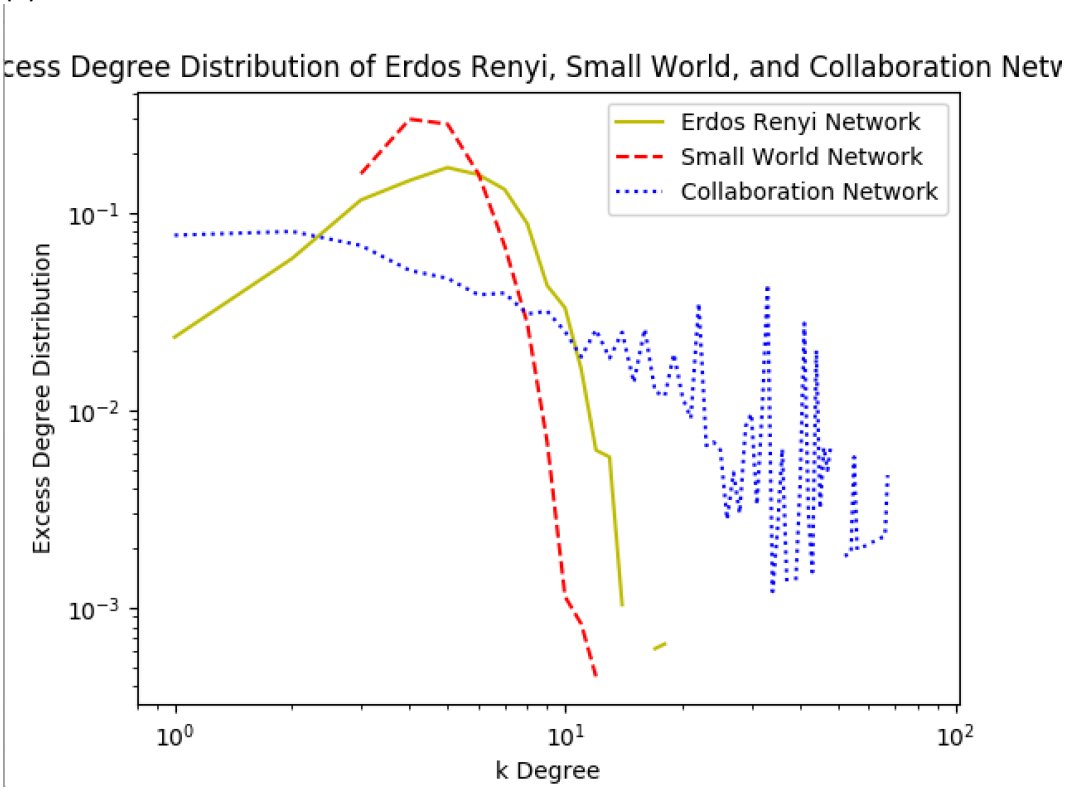


Erdos Renyi network degree follows a closer to binominal distribution; Small world network node degree range is smallest between, the proportion of nodes peaks and then goes down very sharply as node degree increases; Collaboration network node degree range is widest from 1 to almost 100, follows an overall trend of proportion of nodes get smaller but with some fluctuations as node degree increases, and the biggest proportion of nodes have node degree of 1.

(a)



Erdos Renyi network excess degree follows a closer to binominal distribution; Small world network excess degree range is smallest, the proportion of nodes peaks around 6 to 7 node degree and then goes down very sharply; Collaboration network excess degree range is widest, and distribution over the excess degree range is relatively uniform compared to other two networks, it still follows an overall trend of proportion of nodes get smaller but with some fluctuations as node degree increases, and the biggest proportion of nodes have excess degree of 1.

Expected Degree for Erdos Renyi: 5.526135
Expected Degree for Small World: 5.526135
Expected Degree for Collaboration Network: 5.526135
Expected Excess Degree for Erdos Renyi: 5.563518
Expected Excess Degree for Small World: 4.804888
Expected Excess Degree for Collaboration Network: 15.870409

(b)

Given $\{p_k\}$, let p_k be the fraction of nodes with degree k. If we follow an edge then we reach nodes of high degree with probability proportional to k.

The distribution is:

$$\frac{kp_k}{\sum_k kp_k} = \frac{kp_k}{\langle k \rangle}$$

The node of excess degree k will have total degree of k + 1, as we need to include the node that we arrived along, thus excess degree has distribution:

$$q_k = \frac{(k+1)\,p_{k+1}}{\langle k \rangle}$$

We can calculate each of q_i, where i = 1,2,…Max degree to get distribution of $\{q_k\}$

Average clustering Coefficient for Erdos Renyi Network: 0.000963
Average clustering Coefficient for Small World Network: 0.284625
Average clustering Coefficient for Collaboration Network: 0.529636

Network that has the largest clustering coefficient is Collaboration network, because as we see Collaboration network has the largest excess degree range, and also has higher proportion of nodes fall into the large side of the excess degree range compared to the other two networks. Intuitively we know that a group of authors knowing each other tend to collaborate more which makes the number of edges between those authors high.

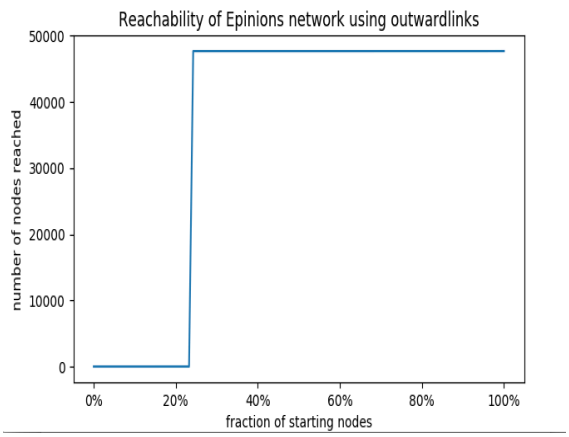
Node 9809 is a part of the ‘OUT’ component because we found out there are 0 nodes intersecting Epinion Max Scc Component and Node 9809 forward link BFS Tree, in the meantime there are 32223 nodes intersecting Epinion Max Scc Component and Node 9809 backward link BFS Tree.

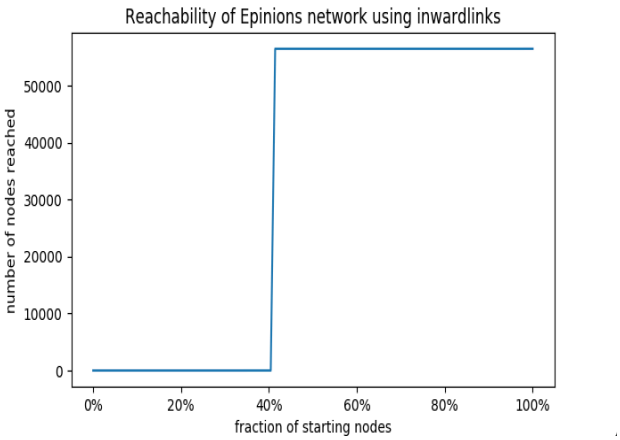
Node 1952 is a part of the ‘IN’ component because we found out that there are 32223 nodes intersecting Epinion Max Scc Component and Node 1952 forward link BFS Tree and there are 0 nodes intersecting Epinion Max Scc Component and Node 1952 backward link BFS Tree.

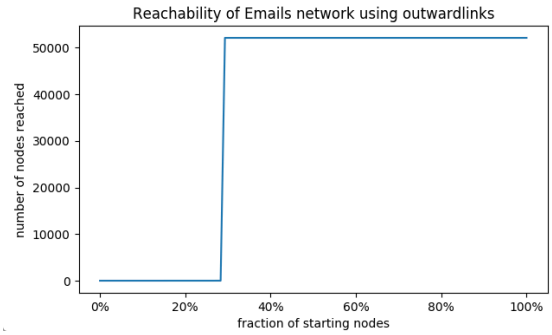
Node 189587 is a part of the SCC component because we found out that there are 34203 nodes intersecting Epinion Max Scc Component and Node 189587 forward link BFS Tree and there are 34203 nodes intersecting Epinion Max Scc Component and Node 189587 backward link BFS Tree.

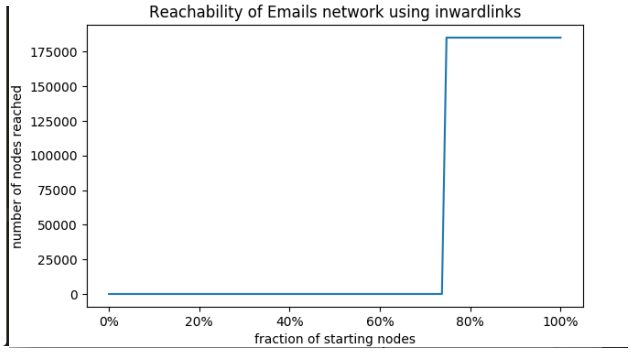
Node 675 is a part of the Out component because we found out there are 0 nodes intersecting Epinion Max Scc Component and Node 675 forward link BFS Tree and there are 34203 nodes intersecting Epinion Max Scc Component and Node 675 backward link BFS Tree.

Question 2.2, Problem Set 1, CS224W









We see all four plots demonstrate' phase by phase' reachability. Starting from small fraction of nodes, the reachability is very small, until the fraction of nodes reached at a certain valuse, the number of nodes reached will jump suddenly and remain flat afterwards.

We see that in both two networks, it takes a higher fraction or nodes to reach high number of nodes using inward links than using outward links.

Emails network: **Disconnected components** have 40382 nodes in total in 15835 different disconnected clusters.
Epinions network: **Disconnected components** have 2 nodes in total in 1 disconnected cluster.
We subtract largest weakly connected component nodes from total nodes, result in the number of disconnected components nodes.

Emails network: **SCC contains** 34203 nodes.
Epinions network: **SCC contains** 32223 nodes.
We just use snap.GetMxScc function to get the SCC component first, then compute the nodes in SCC.

Emails network **:Out component** contains 17900 nodes.
Epinions network: **Out component** contains 15453 nodes
We select a random node in SCC, perform outward direction BFS search, count how many nodes it reached, and then subtract the number of SCC nodes, resulting in the number of Out component nodes.

Emails network **: In component** contains 151023 nodes.
Epinions network **: In component** contains 24236 nodes.
We select a random node in SCC, perform inward direction BFS search, count how many nodes it reached, and then subtract the number of SCC nodes, resulting in the number of In component nodes.

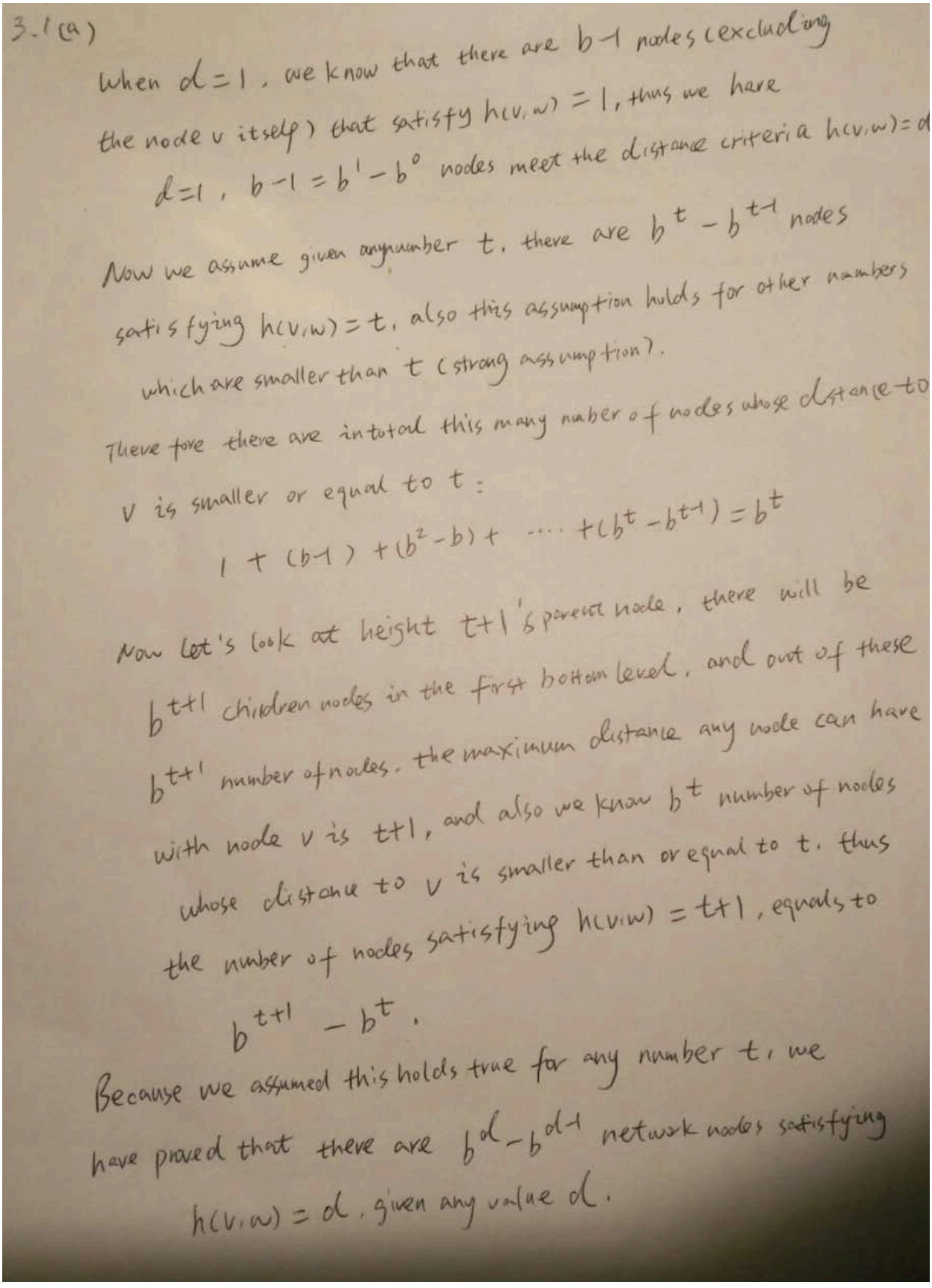
Emails network**: Tendrills** contains 21706 nodes.
Epinions network**: Tendrills** contains 3965 nodes.
The number of nodes in the largest weakly connect component minus number of nodes in SCC, minus number of nodes in In component, minus number of nodes in Out component, resulting in the number of nodes in Tendrills.

In the 1st simulation we randomly select 1000 pairs from the entire graph. Out of 1000 random pairs in Epinions network, 460 of them are connected, probability is 46%. Out of 1000 random pairs in Emails network, 136 of them are connected, probability is 13.6%.

In the 2nd simulation we randomly select 1000 pairs from the weakly connected component. Out of 1000 random pairs in Epinions network, 444 of them are connected, probability is 44.4%. Out of 1000 random pairs in Emails network, 176 of them are connected, probability is 17.6%.

We see that the connected probability in Emails is much smaller than Epinions, possibly due to that Emails network can be think of as a small world where users form small group clusters; also we have seen that the number of disconnected and tendrils nodes are much more in Emails network than Epinions, which also explains why the connection probability is lower in Emails network. On the other hand, sampling from all graph nodes and sampling from weakly connected component don't produce too different result, this is probably due to the weakly connected nodes component is the largest component in these two networks as we see in the last questions' statistic counts.

(a)



3.1(a)

When $d=1$, we know that there are $b-1$ nodes (excluding the node v itself) that satisfy $h(v,w)=1$, thus we have $d=1$, $b-1=b^1-b^0$ nodes meet the distance criteria $h(v,w)=d$

Now we assume given any number t, there are b^t-b^{t-1} nodes satisfying $h(v,w)=t$, also this assumption holds for other numbers which are smaller than t (strong assumption).

Therefore there are in total this many nabers of nodes whose distance to v is smaller or equal to t:

$$1+(b-1)+(b^2-b)+\cdots+(b^t-b^{t-1})=b^t$$

Now let's look at height $t+1$'s parent node, there will be b^{t+1} children nodes in the first bottom level, and out of these b^{t+1} number of nodes, the maximum distance any node can have with node v is $t+1$, and also we know b^t number of nodes whose distance to v is smaller than or equal to t. thus the number of nodes satisfying $h(v,w)=t+1$, equals to

$$b^{t+1}-b^t.$$

Because we assumed this holds true for any number t, we have proved that there are b^d-b^{d-1} network nodes satisfying $h(v,w)=d$, given any value d.

(b)

Say the network has a height of d, from the first problem, we know there will be $b^t - b^{t-1}$ nodes satisfying $h(v,w) = t$, such as there are $b-1$ nodes whose distance to v is 1, $b^2 - b$ nodes whose distance to v is 2, etc.

Therefore for Z which equals to $\sum_{w \neq v} b^{-h(v,w)}$, we ~~can expand~~ can expend Z like below:

$$Z = (b-1) * b^{-1} + (b^2 - b) * b^{-2} + \cdots + (b^d - b^{d-1}) * b^{-d}$$

$$= d - \frac{d}{b} = d(1 - \frac{1}{b}) = \log_b N (1 - \frac{1}{b}) \leq \log_b N \cdot 1$$

Because $b \geq 1$,

So we have $Z \leq \log_b N$, and when $b = 1$, $Z = \log_b N$.

(c)

There are b^{h-1} nodes in tree T' that node v has a probability to create an edge to, and any of such edges is created will be counted in to part of the probability that e pointing to T'.

We have known from the problem that: $P_v(w) = \frac{1}{Z} b^{-h}$

Thus $P_{(e \text{ pointing to } T')} = \text{number of nodes in } T' * P_v(w)$

$$= b^{h-1} * \frac{1}{Z} b^{-h}$$

$$= \frac{b^h}{b} * \frac{1}{Z} * \frac{1}{b^h}$$

$$= \frac{N}{b} * \frac{1}{Z} * \frac{1}{N} = \frac{1}{bZ}$$

And we have proved that $Z \leq \log_b N$.

therefor $P = \frac{1}{bZ} \geq \frac{1}{Z \log_b N}$

We have proved the probability of e pointing to T' is at least $\frac{1}{b \log_b N}$

3-1(d) From the last question, we know that the probability of node v connecting to T' is at least $\frac{1}{b\log_b N}$.

So in a situation of out degree k, none of them connects to T' is ~~equal to~~:

at most: $P_{(\text{not connected})} = \left(1 - \frac{1}{b\log_b N}\right)^k$, where $k = c\cdot(\log_b N)^2$ in our problem set.

We represent $\log_b N$ as y, so we have

$$P_{(\text{not connected at most})} = \left(1 - \frac{1}{b\log_b N}\right)^{c(\log_b N)^2} = \left(1 - \frac{1}{by}\right)^{cy^2}$$

$$= \left[\left(1 - \frac{1}{by}\right)^{by}\right]^{\frac{cy}{b}}$$

$\because\ 1 + x \le e^x$

$$\therefore \left[\left(1 - \frac{1}{by}\right)^{by}\right]^{\frac{cy}{b}} \le \left(e^{-by}\right)^{\frac{cy}{b}} = e^{-cy^2} = \cancel{\log_N e^{cy^2}}$$

$$= N^{-\log_N e^{cy^2}}$$

Let's assign θ equal to $+\log_N e^{cy^2}$,

thus we have $N^{-\log_N e^{cy^2}} = N^{-\theta}$

where $\theta = \log_N e^{cy^2} = cy^2 \log_N e = c\cdot(\log_b N)^2 \cdot \log_N e$

$$= c\cdot\frac{(\log_b N)^2}{\ln N} = \frac{k}{\ln N}$$

Above proves that v has no edge pointing to T' is at most $N^{-\theta}$ where $\theta = \frac{k}{\ln N}$, and k equal to $c(\log_b N)^2$.

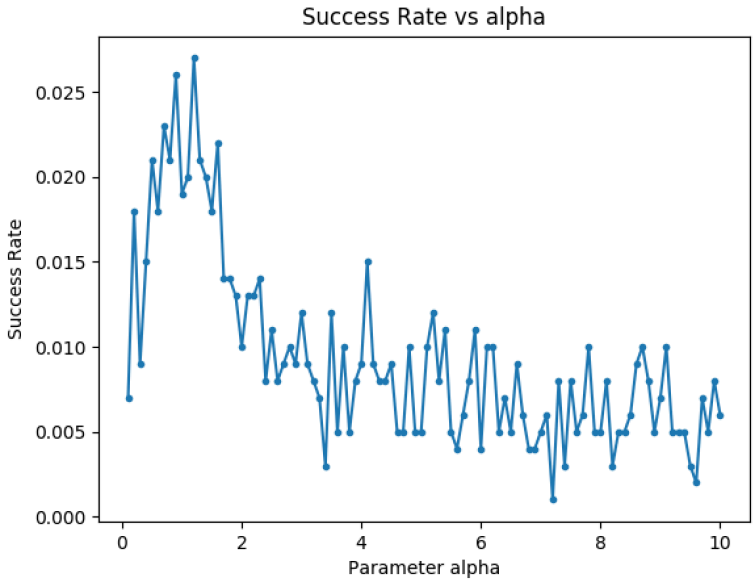
(d)

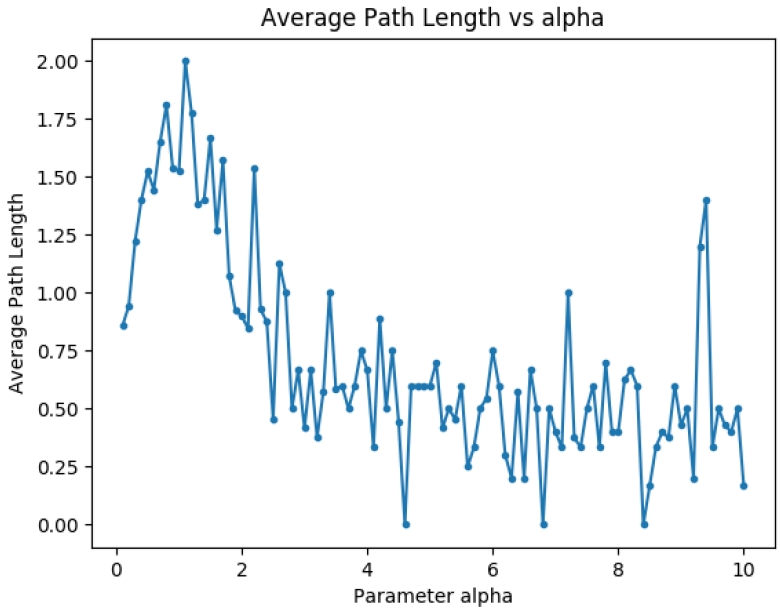
Continue on (d) .

We have seen the probability of not having any edges connecting from ***e*** to tree ***T'*** can be very small as ***N*** increase, and there for there is a high probability that such edge exists for node e to connect with any ***not-t*** node ***u in*** the sub tree ***T'*** , we know that because node ***u*** is in the same subtree with node ***t,*** the distance between node ***u*** to ***t h(u,t)*** will be sure smaller than the distance between node ***v*** and node ***t h(v,t).***

(e)

Suppose node e and node ***t*** are furthest apart, and including those two there are ***N*** nodes in total at leaf node level. In decentralized search at each step the search will jump over at least b number of nodes to move closer to target node ***t,*** because those skipped ***b*** number of nodes belong to the same direct parent node with the current leaf node and their distance to ***target t is*** the same, so that's why those ***b*** nodes will be skipped in each step; as there are ***N*** nodes in total apart from ***e*** and ***t,*** and each search step will jump over ***b*** nodes, so in logbN steps ***e*** can reach ***t,*** thus proving the time is in ***O(logbN)*** steps.





We see that the success rate and average path length follows very similar trend as the parameter alpha value increases, the correlation can be explained that more edges are created, it is quicker to reach to target node. Both the success rate and average path peaks around when alpha is equal to 1, which can be probably explained in this way: when alpha is too large, creating long jump edges is less likely, and if alpha too small, more short jump edges are created, in either of these two scenario it takes more step to complete decentralized search.

Information sheet CS224W:

Analysis of Networks

Assignment Submission Fill in and include this information sheet with each of your assign- ments. This page should be the last page of your submission. Assignments are due at 11:59pm and are always due on a Thursday. All students (SCPD and non-SCPD) must submit their homeworks via GradeScope (http://www.gradescope.com). Students can typeset or scan their homeworks. Make sure that you answer each (sub-)question on a separate page. That is, one answer per page regardless of the answer length. Students also need to upload their code at http://snap.stanford.edu/submit. Put all the code for a single question into a single file and upload it. Please do not put any code in your GradeScope submissions.

Late Homework Policy Each student will have a total of two free late periods. Homeworks are due on Thursdays at 11:59pm PDT and one late period expires on the following Monday at 11:59pm PDT. Only one late period may be used for an assignment. Any homework received after 11:59pm PDT on the Monday following the homework due date will receive no credit. Once these late periods are exhausted, any assignments turned in late will receive no credit.

Honor Code We strongly encourage students to form study groups. Students may discuss and work on homework problems in groups. However, each student must write down their solutions independently i.e., each student must understand the solution well enough in order to reconstruct it by him/herself. Students should clearly mention the names of all the other students who were part of their discussion group. Using code or solutions obtained from the web (github/google/previous year solutions etc.) is considered an honor code violation. We check all the submissions for plagia- rism. We take the honor code very seriously and expect students to do the same.

Your name: Dingchao Zhang
Email: Dingchao@stanford.edu **SUID: dingchao**

Discussion Group:

I acknowledge and accept the Honor Code.

(Signed)