CS-E5740 Complex Networks, Answers to exercise set 4

Sara Cabodi, Student number: 784287

October 14, 2019

Compile with `pdflatex ex_template.tex`

Problem 1

a) Given the assumption that large and sparse ER graphs are tree-like, we use the idea of branching processes and the concept of excess degree to calculate the expected number of nodes at d steps away, n_d, from a randomly selected node in an ER network as a function of $\langle k \rangle$ and d.
The expected excess degree of a randomly selected node is given by the expected number of neighbours (obtained through the friends paradox formula) minus the link followed to get to the random node:

$$\langle q \rangle = \frac{\langle k^2 \rangle}{\langle k \rangle} - 1$$

Since the degree distribution of an ER network is a Poisson distribution, we can exploit the property that the variance is equal to the mean

$$\sigma^2 = \langle x^2 \rangle - \langle x \rangle^2$$

$$\mu = \langle x \rangle$$
$$\sigma^2 = \mu$$
$$\langle x^2 \rangle - \langle x \rangle^2 = \langle x \rangle$$
$$\langle x^2 \rangle = \langle x \rangle^2 + \langle x \rangle$$

If we consider our variable $x = k$ we can then say that

$$\langle k^2 \rangle = \langle k \rangle^2 + \langle k \rangle$$

Replacing $\langle k^2 \rangle$ in the expected excess degree formula, we can define it as

$$\langle q \rangle = \frac{\langle k \rangle^2 + \langle k \rangle}{\langle k \rangle} - 1 = \langle k \rangle + 1 - 1 = \langle k \rangle$$

At each step the number of nodes is multiplied by the expected excess degree, so when computing the number of nodes at d steps away, n_d is:

$$n_d = \langle q \rangle^d = \langle k \rangle^d$$

The giant component appears when $\langle q \rangle > 1$, i.e. $\langle k \rangle > 1$.

b) Results obtained from python code are shown in the next pages. Here are some comments on them. Comments on experimental vs. theoretical calculations and how size affects calculations:

- **k = 0.5**
  In both sizes (10K and 100K) experimental n_d is very close to the theoretical value, The real difference is in the fact that we observe a very low probability of experimentally finding nodes at distance in the high range (13 to 15): though large networks have a higher chance, this not always happens. In the plots, I could observe up to 10 for the 10k network and 13 for the 100k.
- **k = 1**
  Both sizes confirm theoretical expectations as n_d is always not far from to the expected value: 1 to 1.18 range for the 10k network, 0.99 to 1.11 for the 100k network. In both cases the 15 distance is observed.
- **k = 2**
  Experimental results confirm theoretical data, with a limit on the network size: n_d cannot obviously reach 10k and 100k (total amount of nodes), respectively.

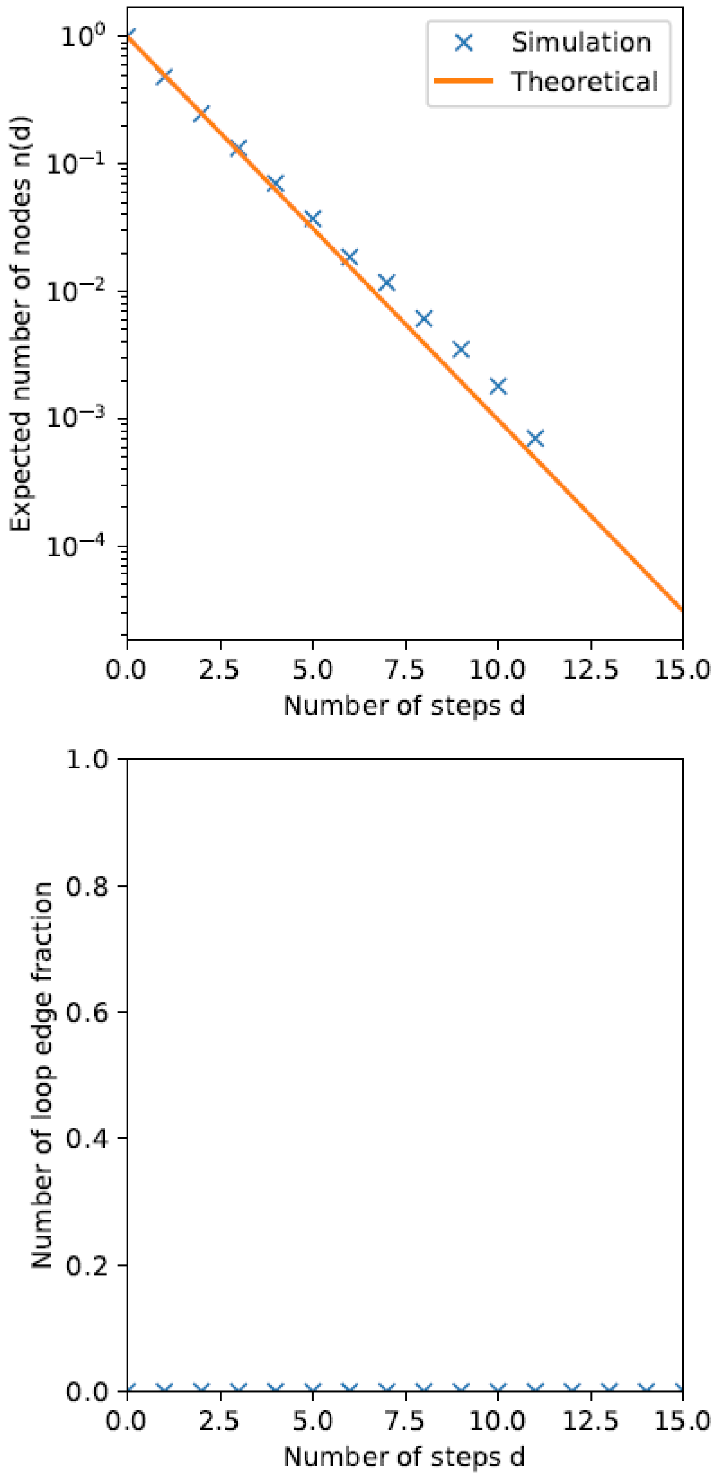


Figure 1: ER network breadthfirst with k=0.5 and network size = 10000

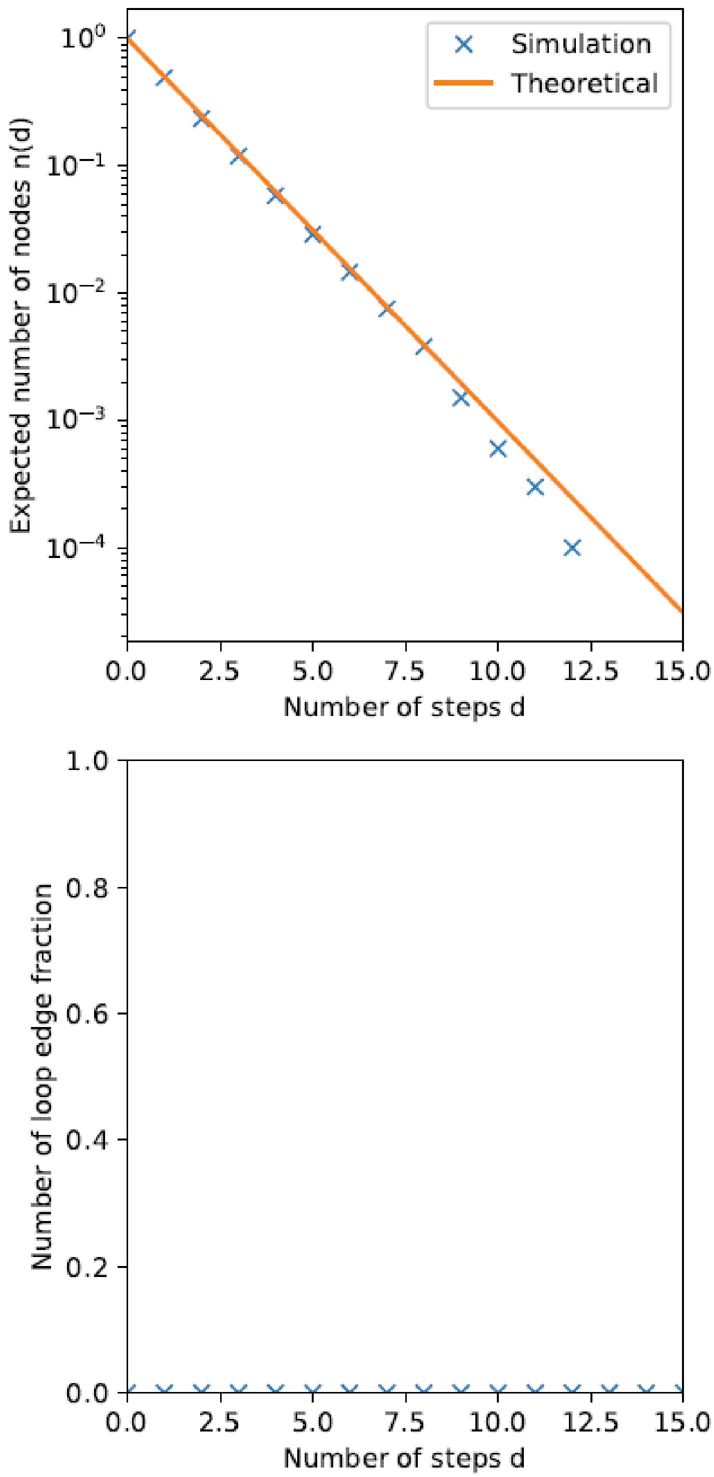


Figure 2: ER network breadthfirst with k=0.5 and network size = 100000

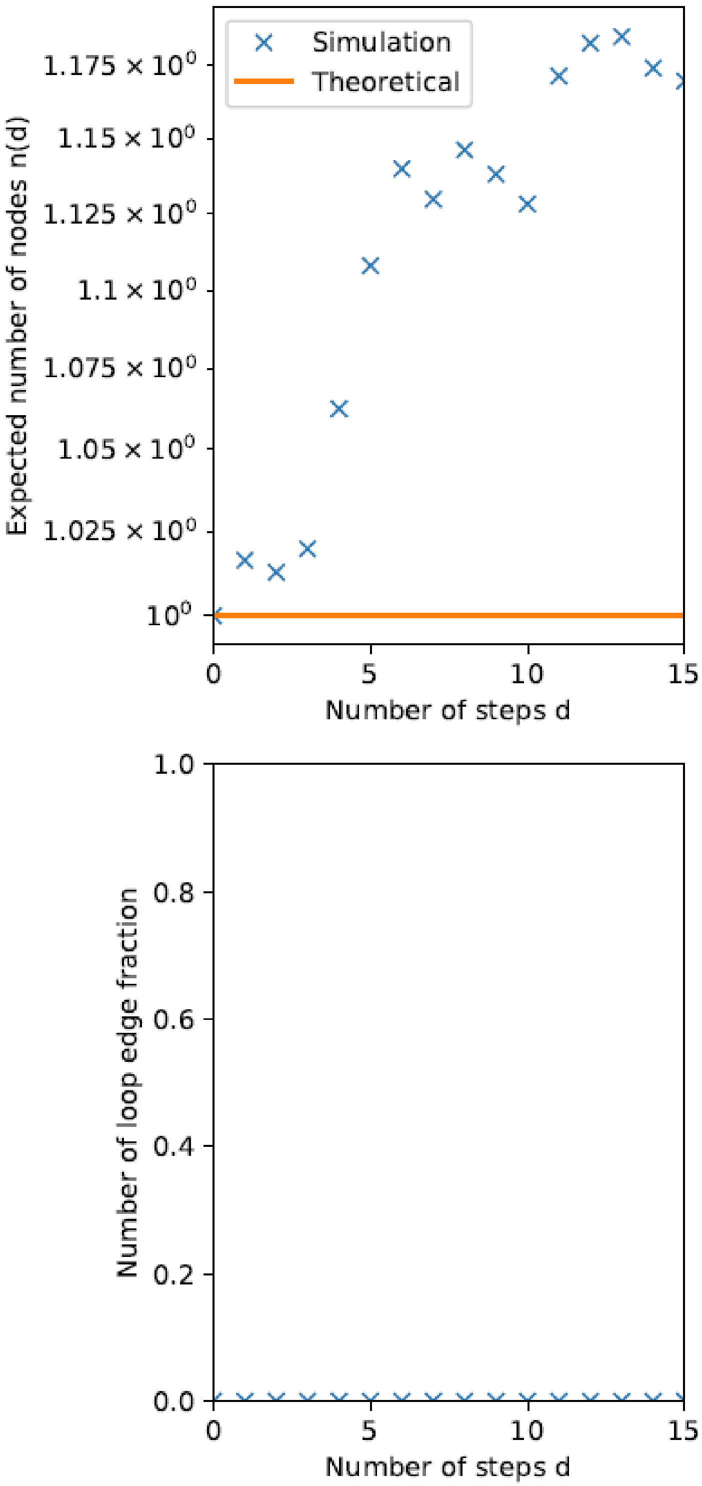


Figure 3: ER network breadthfirst with k=1 and network size = 10000

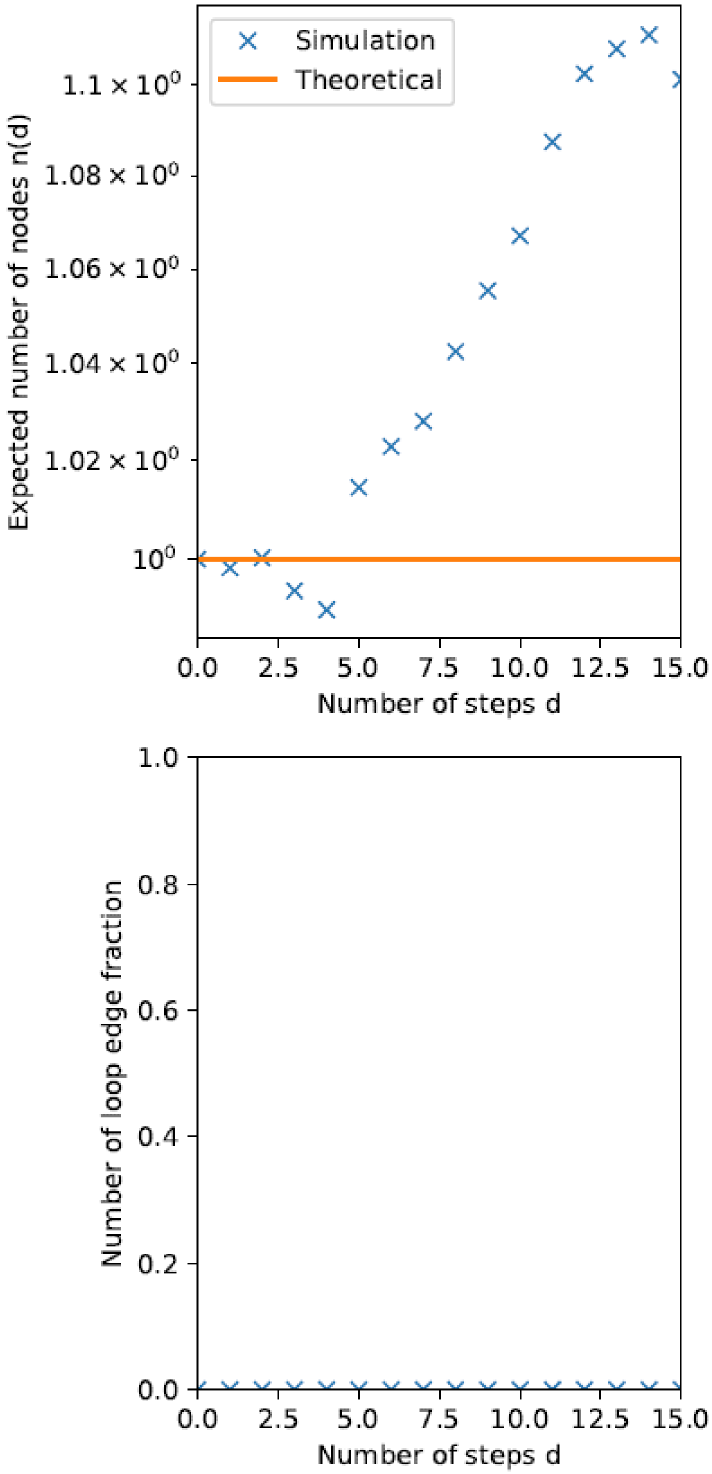


Figure 4: ER network breadthfirst with k=1 and network size = 100000

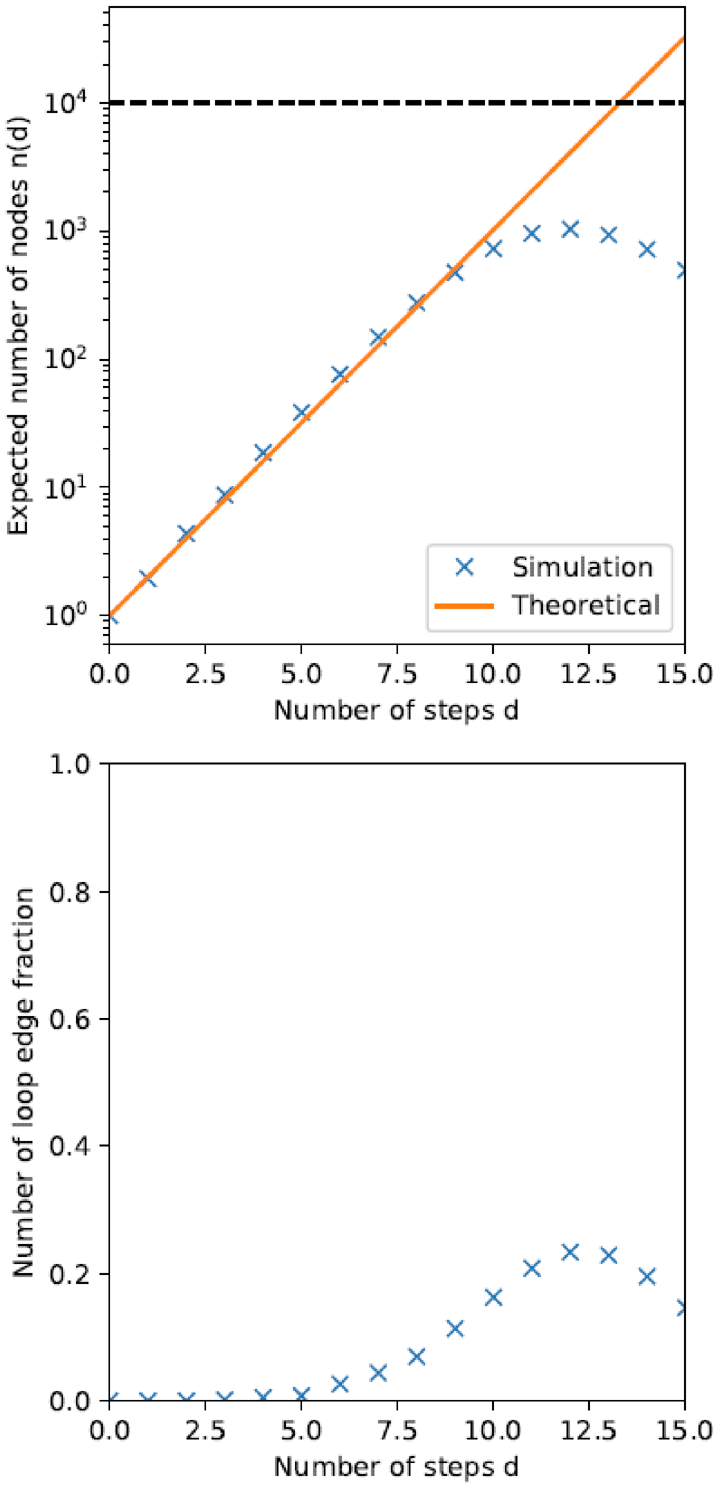


Figure 5: ER network breadthfirst with k=2 and network size = 10000

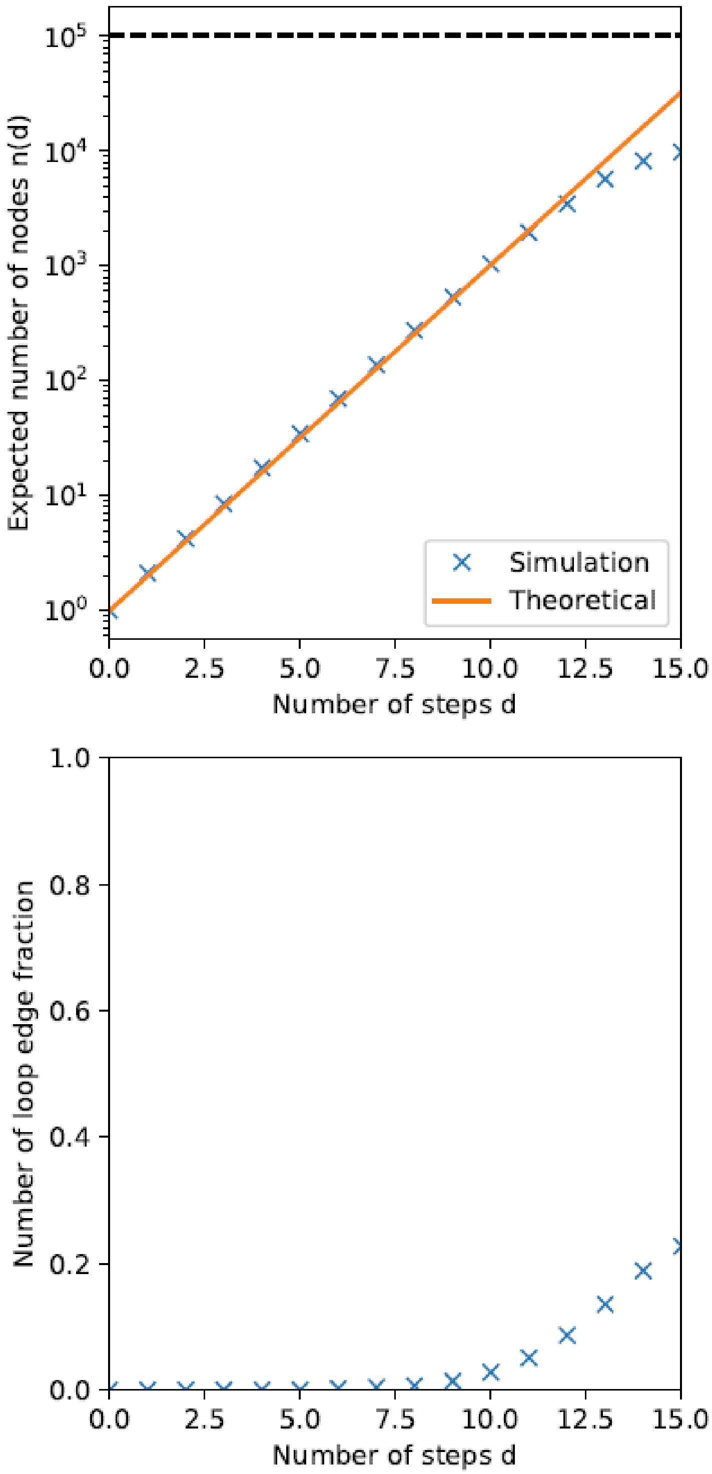


Figure 6: ER network breadthfirst with k=2 and network size = 100000

c) When $\langle k \rangle = 0.5$ or $\langle k \rangle = 1$ the assumption of tree-likeness is confirmed, since there aren't visible cases of loop edges between nodes. Instead, when $\langle k \rangle = 2$, loop edges increase with the number of steps d. In particular, loops start to appear around $d = 5$ in the case of $N = 10k$ and around $d = 8.5$ when $N = 100k$.

   As $\langle k \rangle = 2$ networks have a giant component, the increasing number of loop edges observed is related to the higher size of n_d: larger sets of nodes correspond to higher values of the loop edge fraction.

   Concerning the impact on the calculation of n_d, loop edges limit the n_d value to be smaller than the expected one, in the $\langle k \rangle = 2$ experiments. The other limit on the n_d value is given by the total amount of nodes (10k or 100k): in the 10k experiments we observe a relatively higher impact of the lower number of nodes, as both experimental n_d values and loop edges reach a maximum around $d = 12$, then decrease. A certain variability with respect to the expected theoretically values could also be related to the non-uniform distribution of degree values and to the stochastic nature of experimental sampling.

d) The plot confirms that the giant component appears from $\langle k \rangle = 1$ on, with a size going from units to nearly 90% of nodes in the network.

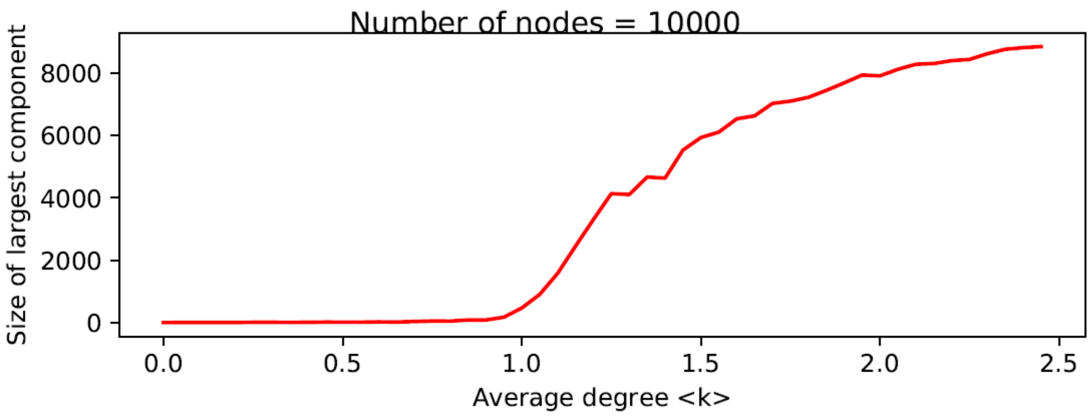


Figure 7: ER network with network size = 10000

e) The curve has a peak at $\langle k \rangle = 1$, as expected since it's the percolation transition point. It shows that adding a single link changes a lot the largest component as it happens when the giant component is formed. So to small changes (around the percolation point) correspond big changes in the size of the largest component.

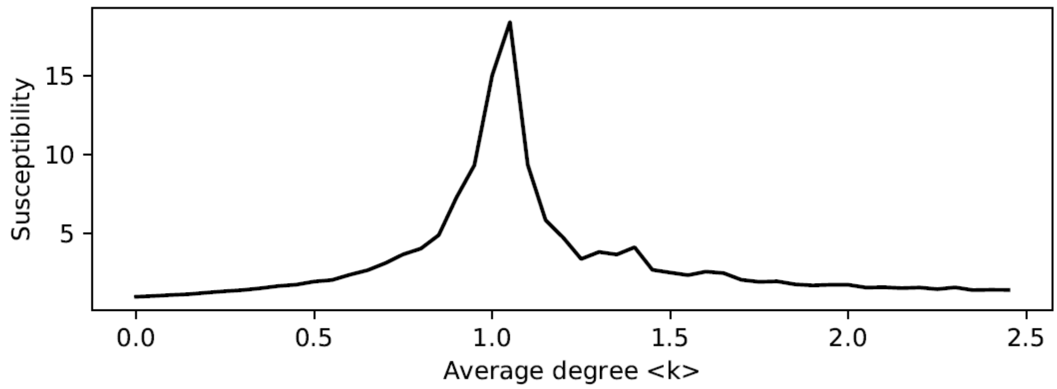
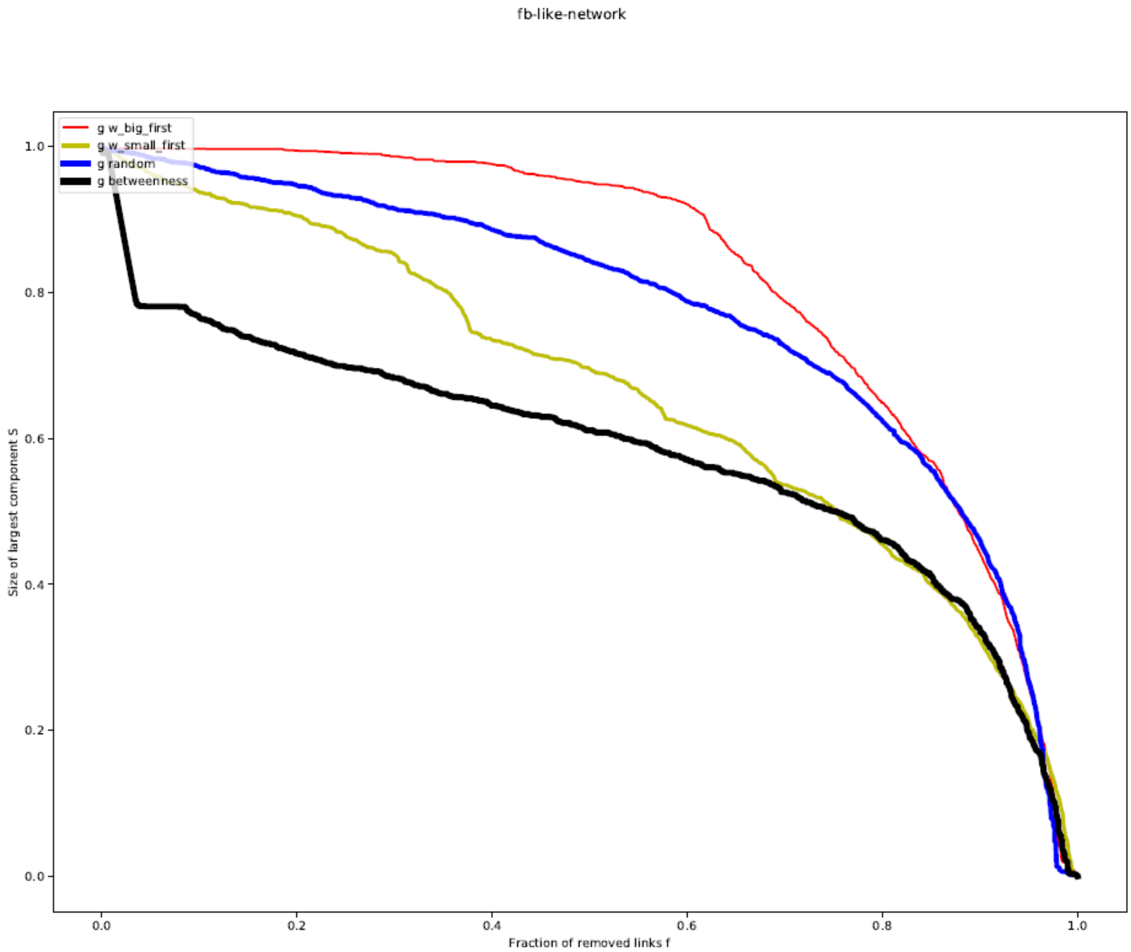


Figure 8: Susceptibility of ER network with network size = 10000

Problem 2

a) Here is the graph representing the size of the largest component S as a function of the fraction of removed links f for the four different orders.



b) The most vulnerable of the approaches is the one where we remove the links based on their betweenness centrality value (descending order): intuitively, this is due to the fact that nodes with high betweenness are critical for many node-to-node paths. The least vulnerable is the one with removal of strong links first (an intuition of the reason for this is given in answer 2c). As the fraction of removed links increases and goes close to 1, it is hard to tell which approach is better/worse than the others, as in all cases the graph tends to break in small components.

c) The weak links appear to be more important for integrity, because they allow the network to keep the largest component untouched for longer. Intuitively, as edge strength/weakness is related to edge weight, a weak edge is one with low traffic. Since this is a social media network, this means that a weak edge is located in lowly active part of the graph, whereas strong links characterise highly active subgraphs, where alternative redundant paths can be found when breaking a link. So overall, in my opinion, removing a weak link has a higher chance of breaking the connected component and to decrease the size of the largest component faster.

d) Removing edges with high betweenness turns out to be even more critical for the integrity of the the core part of the giant component. This is probably due to the fact that high betweenness edges are part of many node-to-node connection paths (independently from traffic related weights). So they have higher chance of breaking the largest component. Instead, the random removal strategy is not biased, so it shows an intermediate level of integrity. This behaviour is visually shown in the plot of point a).