Social Network Analysis HW1

Hamid Nemati

File1 has code that needed to be written with **NetwrokX** package which is available for the WINDOWS environment but it's slow. File2 has the same thing done with **Graph-tool** package which is faster but only available for Linux and Mac; if we want to run File2 in windows we need ducker to do that or we can use google colab which runs on Linux. I chose to use google colab and a virtual machine with Linux environment. (Both .html and .ipynb are in the files. The code is very clean so few comments were needed)

As Erdős–Rényi article (1959) about Poisson Random Network states when the probability *p* goes beyond the threshold $\boldsymbol{t(n) = \frac{\log(N)}{N}}$, the network becomes path connected. It means when the expected number of connections, $E[k] = p \times (n-1)$, is equal or more than $Log(N)$; the probability of a node being isolated goes to zero as N goes to infinity.

We want to show, mean geodesic distance between pairs of nodes, $\langle d \rangle \; or \; d(n)$, is small relative to the total number of nodes: $d(n) = O(logN) \; as \; N \to \infty$

$$\langle d \rangle \geq (1+\varepsilon) Log(N), \qquad \varepsilon > 0$$

$$\frac{\langle d \rangle}{N} \to 0$$

$$\frac{\langle d \rangle}{\frac{\log N}{\log \langle k \rangle}} \to 1$$

Let's first look at the mathematical proof. We imagine instead of a random network we have a Cayley tree (each node has a degree k). In step 1 we can reach k node. In step 2 we can reach $k(k-1)$ and so on.

$$\frac{k((k-1)^l - 1)}{k-2} \sim k^l$$

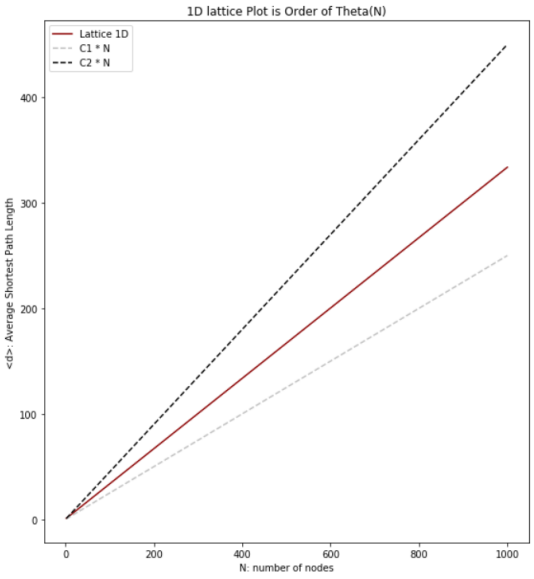
So if we want to reach $n-1$ node we approximately need $k^l = n$. If we take a log of both sides we are going to have:

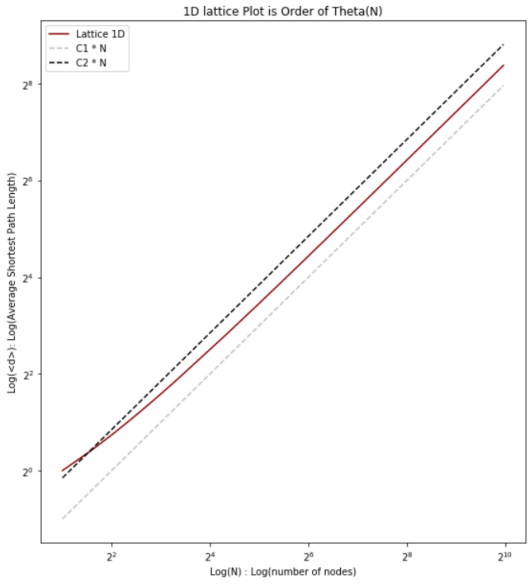
$$l = \frac{\log n}{\log k}$$

So if we have a regular tree with an average degree $\langle k \rangle$, with $\frac{\log n}{\log k}$ steps we can reach every node. And we can expand this to random graphs because most of the links are reaching new nodes as the number of unreached nodes is far larger than reached nodes (we can actually prove this with Chernoff Bounds which states: if X is binomial variable then: $\boldsymbol{P\left(\frac{E[X]}{3} \leq X \leq 3[E]\right) \geq 1 - e^{-E[X]}}$)

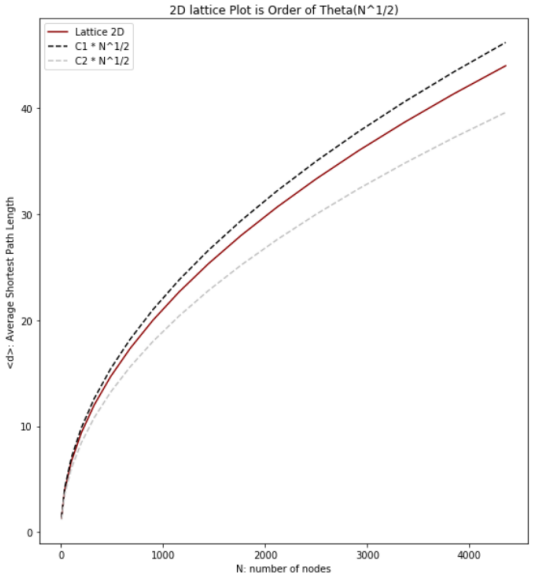
Now let's take a look at the simulation and interoperate the plots.

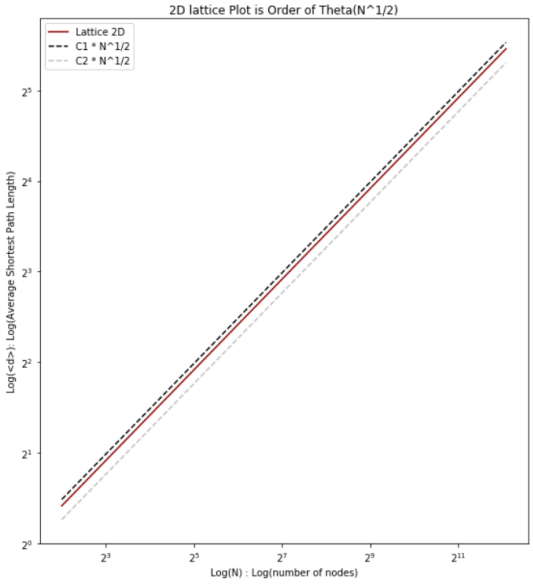
1. One Dimensional Lattice: The dashed lines are Order of *N* with coefficients of C1 and C2
    a. Growth Order: $\theta(N)$
    b. Time = 10 Second



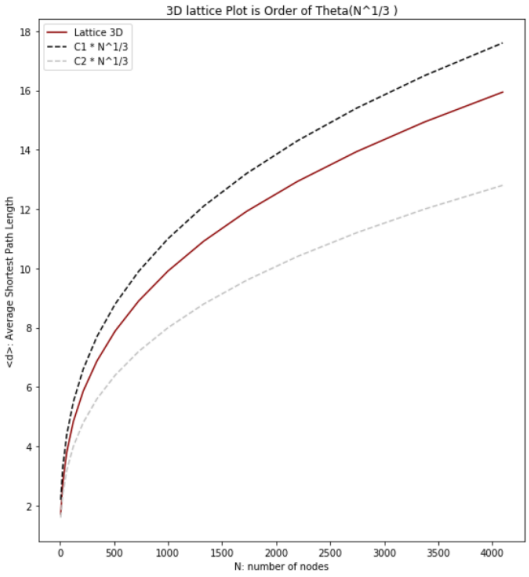


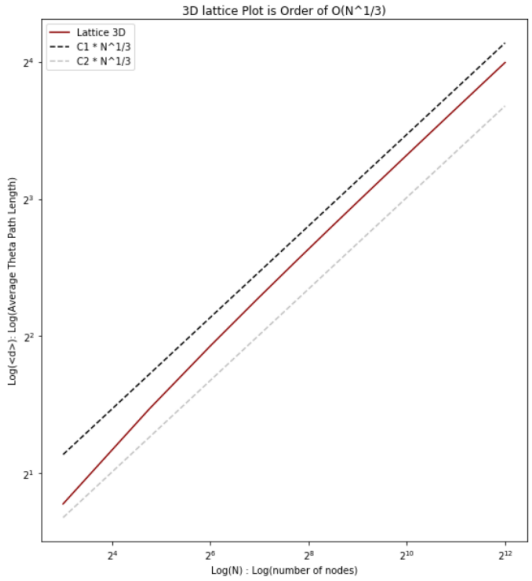
2. Two Dimensional Lattice: The dashed lines are Order of $N^{\frac{1}{2}}$ with coefficients of C1 and C2
    a. Growth Order: $\theta(N^{\frac{1}{2}})$
    b. Time: 270 Second



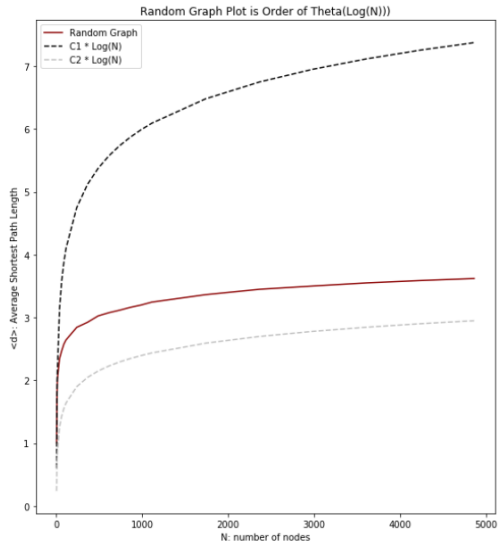


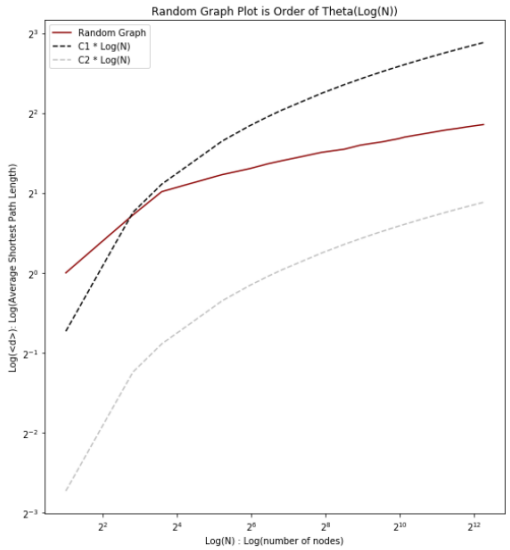
3. Two Dimensional Lattice: The dashed lines are Order of $N^{\frac{1}{3}}$ with coefficients of C1 and C2
    a. Growth Order: $\theta(N^{\frac{1}{3}})$
    b. Time: 210 Second



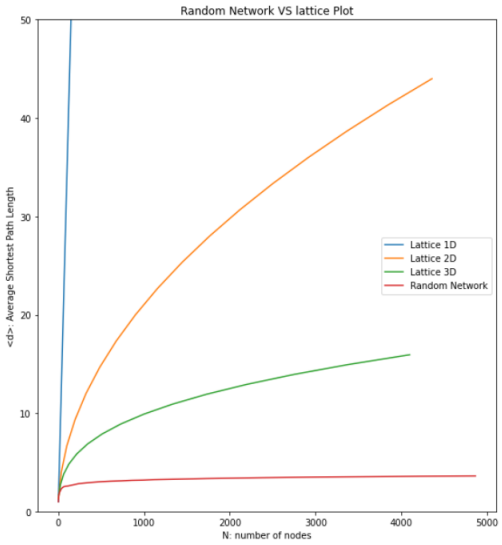


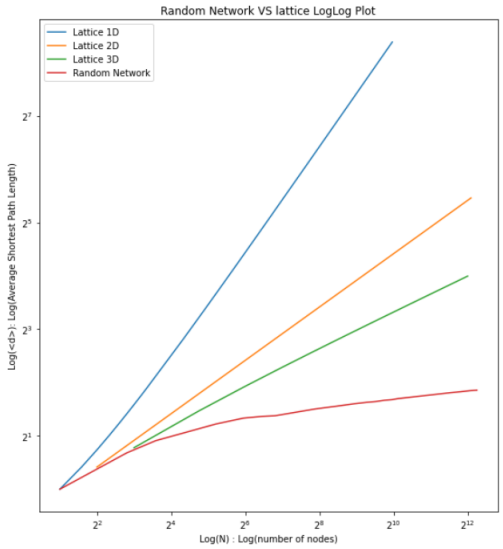
4. Erdos-Renyi Random Network: The dashed lines are Order of $\log N$ with coefficients of C1 and C2
    a. Growth Order: $\theta(\log N)$
    b. Time for 10 rounds: 5686 Second # Note about the value of **S**: We need to run the random Network multiple times because it gives us a different shape each time and each shape has a different Average-Path-Length, So taking an average here gives a better value, and the bigger it is the better but due to computational limitation we used 10.
    c. Average Time of Each round: 568 Second
    d. $\boldsymbol{p = \frac{1.5 * \log N}{n}}$ #Note about the value of **P**: We multiplied the probability suggested by Erdos-Renyi by $\mathbf{1.5}$; This is because the base probability itself will not make the graph connected unless the number of nodes goes to infinity. The more connected the network gets the less Average-Path-Length gets. That's why we can see the line is not fitted very well between its pairs.
    e. Producing an E-R Random Network is in growth order of $O(N^2)$
    f. # Note about the value of **N**: we can clearly see what Erdos-Renyi proved can only be true if N goes to infinity because only then the probability of having a connected graph goes to 1. In smaller N we can see the line is not close to $\log N$ at all.





5. All together
    a. We can see in the left plot that Average Distance is very close to a flat line and when we draw the loglog plot we see that unlike the lattices it's not increasing linearly and it has a very small growth rate.





6. If we use google colab and graph-tools library which due to its C++ core is much faster we can do all the steps above in a shorter amount of time (about 12 minute). The graph-tools poor documentation I wasn't able to fine an *is_conneced()* function and because of that the value calculated for average shortest path length was wrong due to the existence of isolated nodes. So I replaced the random part with NetworkX's code that's why we see the value for time is larger than I said previously. You can view the code in the link below:
https://colab.research.google.com/drive/1d8cfTx1sPReVXWufIB9vizipgoOfvtbn?usp=sharing