Part 1

We used the library **networkx** to create and manage the p-graph and er-graph, togheter with **numpy** and **seaborn** in order to manage matrices and plots respectively.

The script we wrote and others material related to this challange can be fount on: GitHub.

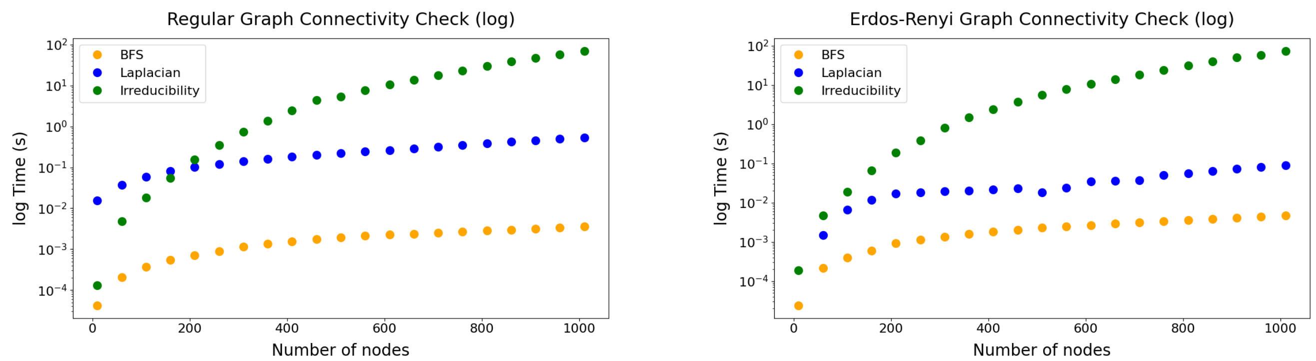

Figure 1: Comparison of methods **BFS, Laplcian and Irreducibility** for **r-regular** and **Erdős–Rényi**

We want use as metric the time for two reason:

- It's easy to compute with the python library **time**
- We think is an important metric for a Data Center

With this choice we found a problem: due to the use of a concurrent system, the library **time** doesn't return the exact time used by the functions. To avoid this and keep the time as metric we chose to use a median. In practise we did a bootstrap: for each k we chosen, we repeat the measure 15 times and then we get the median.

About Erdős–Rényi we set the probability $p = \lambda(\log(n)/n)^{MIT}$ using $\lambda = 1$ in order balance the connected and not connected graphs.

About the Laplacian method, in order to save time and not compute all the eigenvalues we used the UL Factorization as suggested here.

After comparison of the three methods, we can conclude the **best is the visit of the graph**.

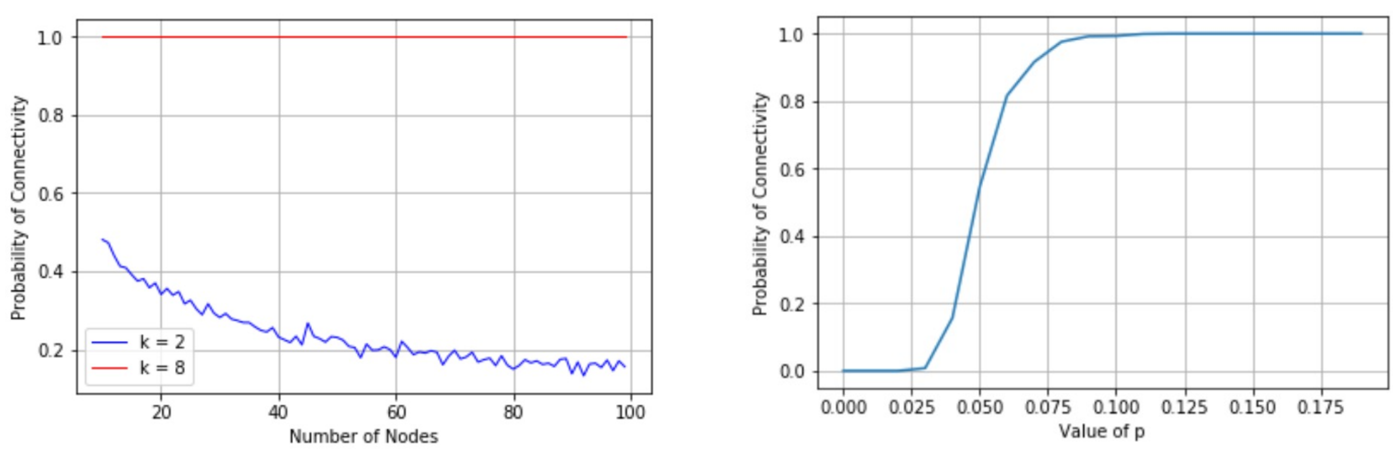

Figure 2: Measure of probability of connectivity

We can notice that for Erdos-Renyi graph, after a value of 0.1 for p, the probability of having a connected graph is equal to one. For the R-regular , we can be sure that for k > 2 the probability of connectivity is always 1.

Part 2

Part 2.1

Below we report on the left the values of N, S and L for the fat tree; while on the right the system to be solved to find r such as to have the same values as the fat tree also in the jellyfish.

$$\begin{cases} N = \frac{n^3}{4} \\ S = \frac{5n^2}{4} \\ L = \frac{3n^3}{4} \end{cases} \qquad \begin{cases} N = S(n-r) \\ L = \frac{Sr}{2} + S(n-r) \end{cases} \implies \begin{cases} S = \frac{5n^2}{4} \\ r = \frac{4n}{5} \end{cases}$$

The floor has been laid out since r must be int, so at the end we have $r = \frac{4n}{5}$.

Part 2.2

We know that the TH bound for an all-to-all r-regular random graph is $TH \leq \frac{3N}{\bar{h}\nu}$, since $\nu = \frac{N(N-1)}{2}$, using the formulas seen previously for S and r we obtain:

$$TH \leq \frac{6}{\bar{h}(N-1)} = \frac{24}{\bar{h}n^3}$$

Part 2.3

For the case of the fat tree we have seen some graphs with reasonable values of n and r and generalized the number of path at fixed length. We are considering only the even n since it is not possible have a fat tree topology with other n. To find $\bar{h}$ for the Fat-Tree we count all possible permutation of path among all passible pair of servers for each fixed length. For the Jellifish we used the formulas in the slides and adding 2 since only the switches are a r-regular graph and we must connect the servers; we are taking the bound for $\bar{h}$. With P_{h_i} we represent the number of path of lengt h_i with $i = \{2, 4, 6\}$.

$$\begin{cases} P_{h_2} = \binom{\frac{n}{2}}{2}\frac{n}{2}n \\ P_{h_4} = \binom{\frac{n}{2}}{2}(\frac{n}{2})^3 n \\ P_{h_6} = \binom{n}{2}(\frac{n}{2})^6 \end{cases} \implies \bar{h}_{Fat-Tree} = \frac{2P_{h_2} + 4P_{h_4} + 6P_{h_6}}{P_{h_2} + P_{h_4} + P_{h_6}}$$

$$\bar{h}_{Jellyfish} = \frac{\sum_{j=1}^{k-1} jr(r-1)^{j-1} + kR}{S-1} + 2 = \frac{\sum_{j=1}^{k-1} j\frac{4n}{5}(\frac{4n}{5}-1)^{j-1} + kR}{\frac{5n^2}{4} - 1} + 2$$

With:

$$k = 1 + \left\lfloor \frac{log(\frac{n^3}{a} - \frac{2(\frac{n^3}{4}-1)}{\frac{4n}{5}})}{log(\frac{4n}{5}-1)} \right\rfloor \qquad \text{and} \qquad R = \frac{5n^2}{4} - 1 - \sum_{j=1}^{k-1} \frac{4n}{5}(\frac{4n}{5}-1)^{j-1}$$

n	N	S	L	$TH_{Fat-Tree}$	$TH_{Jellyfish}$
10	250	125	750	$4.03 \cdot 10^{-3}$	$5.43 \cdot 10^{-3}$
20	2000	500	6000	$5.00 \cdot 10^{-4}$	$6.70 \cdot 10^{-4}$
30	6750	1125	20250	$1.50 \cdot 10^{-4}$	$2.00 \cdot 10^{-4}$
40	16000	2000	48000	$6 \cdot 10^{-5}$	$8 \cdot 10^{-5}$
50	31250	3125	93750	$3 \cdot 10^{-5}$	$4 \cdot 10^{-5}$