Part 1

To generate graphs and work on them we uses the functions provided by **networkx** library, in particular the functions for BFS and the algebraic connectivity method are provided by it, whereas we implemented the irreducibly method by our-self, exploiting the functions in the **numpy** library. Finally, **seaborn** and **matplot** return the graphs.

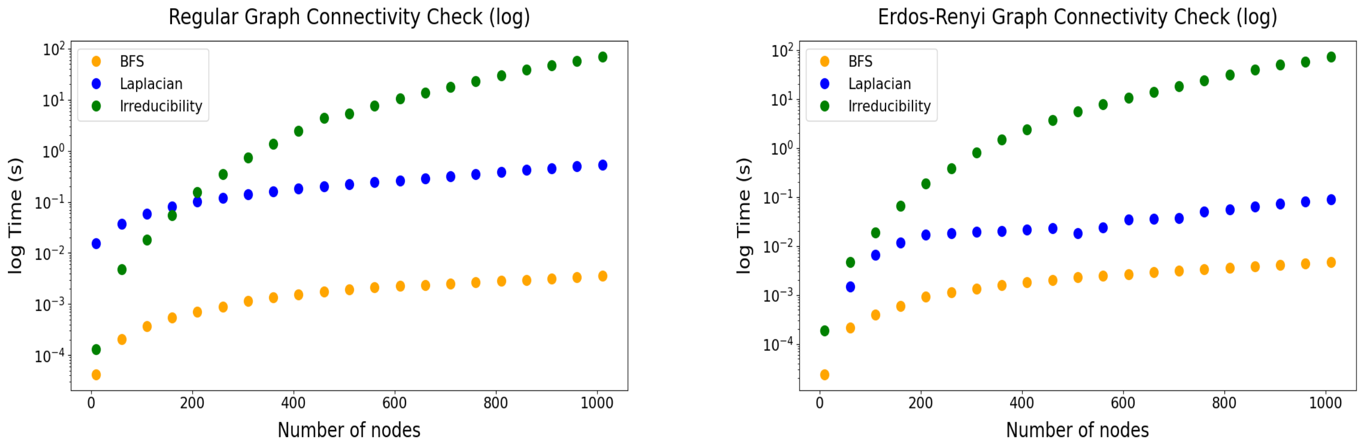

Figure 1: Comparison of methods **BFS, Laplcian and Irreducibility** for **r-regular** and **Erdős–Rényi**

Time has been chosen as metric because it is easy to compute and it is an important metric to evaluate the performance in a Data Center. However, due to the use of a concurrent system, the time we obtain is not the exact one needed to run the functions. To obtain a more reliable result we use the **bootstrap**: for each given k (number of nodes), we repeat each connectivity test 15 times and then we get the **median** of the results of each method. To obtain a better idea of the **walk of the curves** we smooth the curves thanks to the Savitzky–Golay filter and we use a **logarithmic scale** for the y-axis. For what concerns the settings we used, the probability used for the Erdős–Rényi ($p = \lambda(\log(n)/n)^{MIT}$) is set using $\lambda = 1$ to balance the connected and disconnected graphs. For **algebraic connectivity method**, to save time (not computing all the eigenvalues) we used the **UL Factorization** as suggested here.

The results show that the **irreducibly method** is the slowest, because of all the operations involving matrices. Instead, **the best is the BFS**, that in the worst case takes $O(k^2)$. The **irreducibly method** is also slow, but thanks to the UL Factorization it speeds up. In addition, we have noticed that the time it takes to be run is **a lot variable** (far away more than the other two methods).

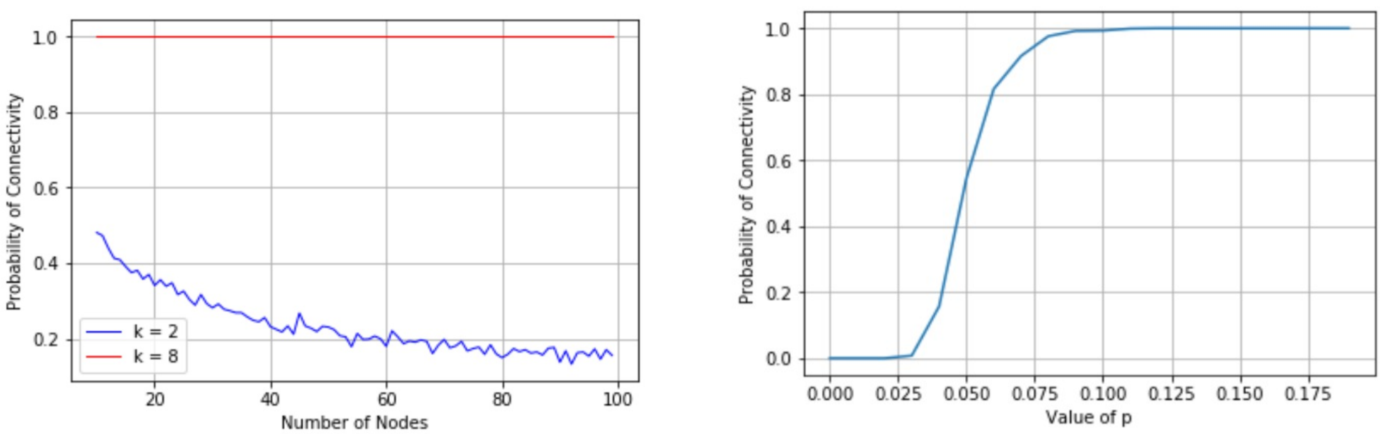

Figure 2: Measure of probability of connectivity

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	$3 \cdot 10^{-5}$	$4 \cdot 10^{-5}$