EE2T21 – Data Communications Networking
Bonus Assignment No. 2

18 June 2024

Kevin van der Toorn
Student number: 4482697

Introduction

The program is written in Python as a Jupyter (iPython) notebook. It is written using Python 3.9.7, but assumed to work on other Python 3.x versions as well (especially higher version numbers). The Python package NetworkX is used, but only to check whether two nodes are disconnected. The other required packages are included in the Anaconda Python distribution. The notebook is available at github.com/KevinvdT/TUDelft_EE2T22_Assignment-2. It can be cloned using Git or downloaded as ZIP file.

Experiment

The time complexity of Dijkstra's algorithm is

$$O(E + V \log V),$$

where E and V are the number of edges and nodes respectively.[1]. For an Erdős Rényi random graph, each pair of nodes has probability p to have an edge connecting them, so $V \approx pE(E-1)$ (on average), meaning

$$O(E + V \log V) = O(E + [pE(E-1)] \log[pE(E-1)]) = O(E^2 \log E^2).$$

Therefore, we expect the results graph of the experiment to (roughly) have the shape of $y = c \cdot x^2 \log x^2$, see Figure 1. The result of the experiment is given in Figure 2. As we can see, the outcome data seems to match the predicted results.

Unfortunately, I was not able to do the experiment for networks of size $N \geq 2^{11}$ because of memory constraints. I wanted to implement the graph, nodes, and edges in code myself instead of using a library. I thought this would teach me the most. In hindsight, it might have been better if would have used a ready-made graph library like NetworkX, because this would have been more memory-efficient and would likely have made networks with size $N \geq 2^{11}$ possible.

[1] Source: en.wikipedia.org/wiki/Dijkstra's_algorithm

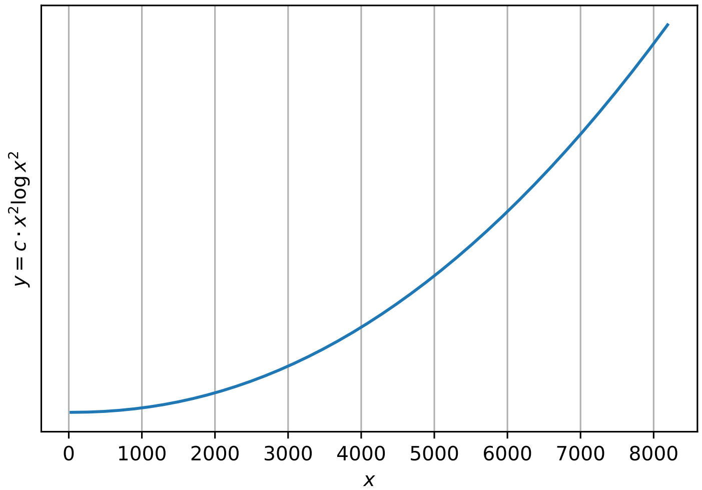

Figure 1: Plot of $y = c \cdot x^2 \log x^2$, with c a proportionality constant

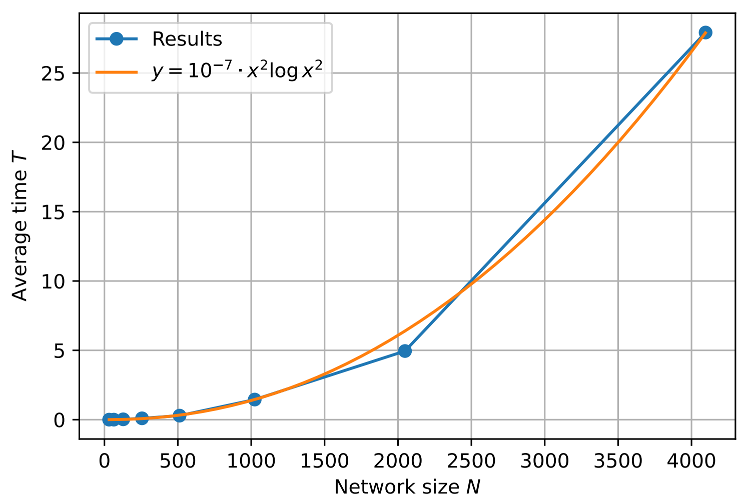

Figure 2: Plot of $y = x^2 \log x^2$