1 Design

In this assignment , we have implemented the simulation of simple worm propagation in a medium-scale network by using discrete-time simulation method. We assume an isolated network with omega(Ω = 100,000) IP address space which means the IP range in our network ranges from 1 to 100,000. It is assumed that there are N = 1,000 computers that are vulnerable to the worm in this isolated network. The IP addresses of the vulnerable computer has the following pattern of IP addresses:

1, 2, 3,..., 10,
1001, 1002, ..., 1010, 2001, 2002, . . . , 2010, ...

From this we can see that, each cluster of 10 computers with the consecutive IPs are vulnerable to the worm, and in every 1000 consecutive IP addresses there will be one cluster of 10 vulnerable computers (so there are 100 clusters of vulnerable computers overall).

We further assume that worm starts to propagate infection within the network initially from 1 machine which has IP address of 1001. The scan rate(n) of infected machine is 3. This implies that a worm-infected computer can scan to 3 other IP addresses in the network at each time step. A vulnerable computer is immediately infected if a scan finds it and this newly infected computer can also begin infection other 3 IP addresses from the next time step. In this way the worm propagates and infect the whole computer in a network.

2 Implementation

We need to find the number of infected computers at each time step t(t=1,2,3...) which is represented as I(t). We simulate the worm propagation 3 times to get the three vector of the number of infected IP, I(t). The simulation ends when all the vulnerable machines are infected. At initial point, I(0) = 1. We have implemented simulation of two kinds of scanning:

2.1 Random Scanning

In this method, an infected computer x randomly selects another 3 IP addresses with in the entire IP address space in a time unit.

2.2 Sequential Scanning

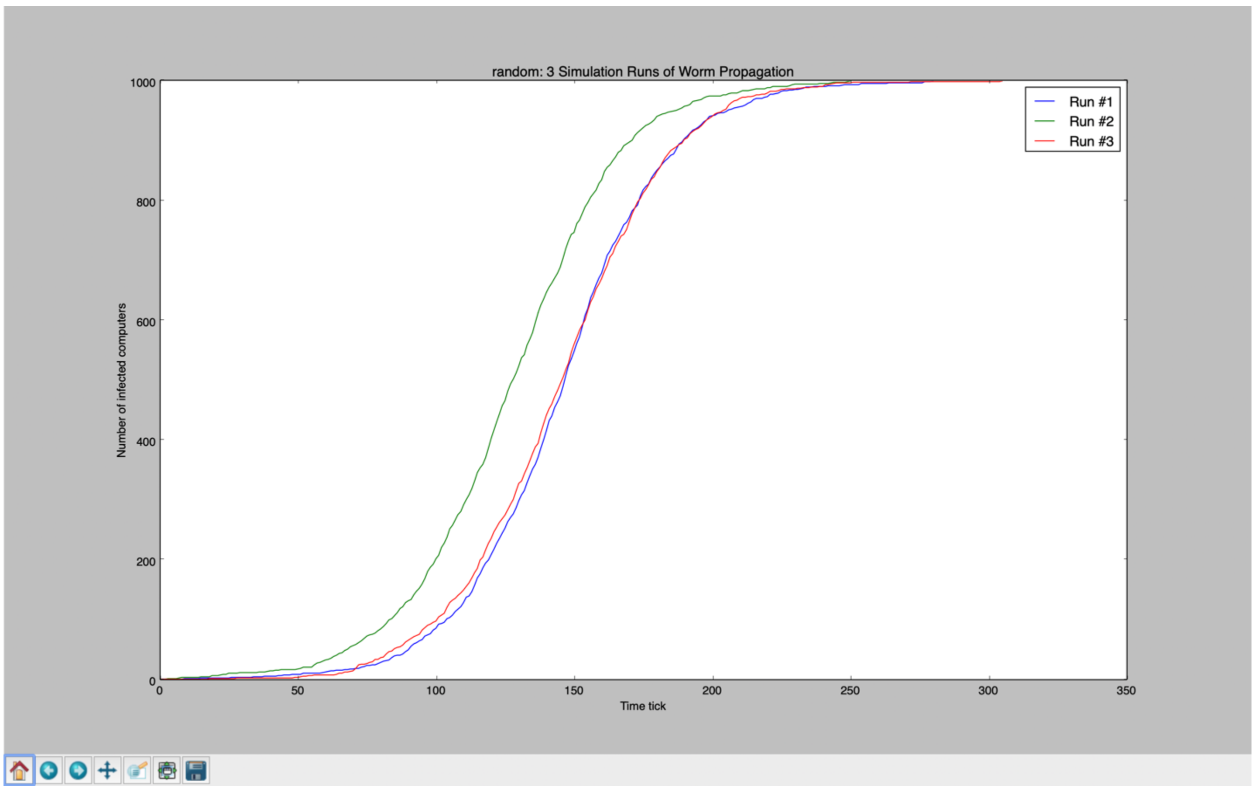
In this method, an infected computer at each time step with IP value x picks the target IP address value y by the following rule:

1. With probability p = 0.8, it picks a random value y such that $y \in [x-10, x+10]$
2. With the remaining probability 0.2, it picks a random value y between 1 to 100,000.

3 Results and Analysis:

We performed three simulation runs for both random scanning and local-preference scanning method. As an output, we plotted a curve that shows the number of infected computers at each time step as shown in the figure1 and 2. The plot of worm propagation across the network follows a sigmoid curve (S-curve) in both the methods. It can be seen that the infected computers at the beginning increases slowly and rapidly increases with the time as the number of infected computers increases and finally reach the maximum point.

Figure 1: Three simulation run on worm propagation using random scanning method



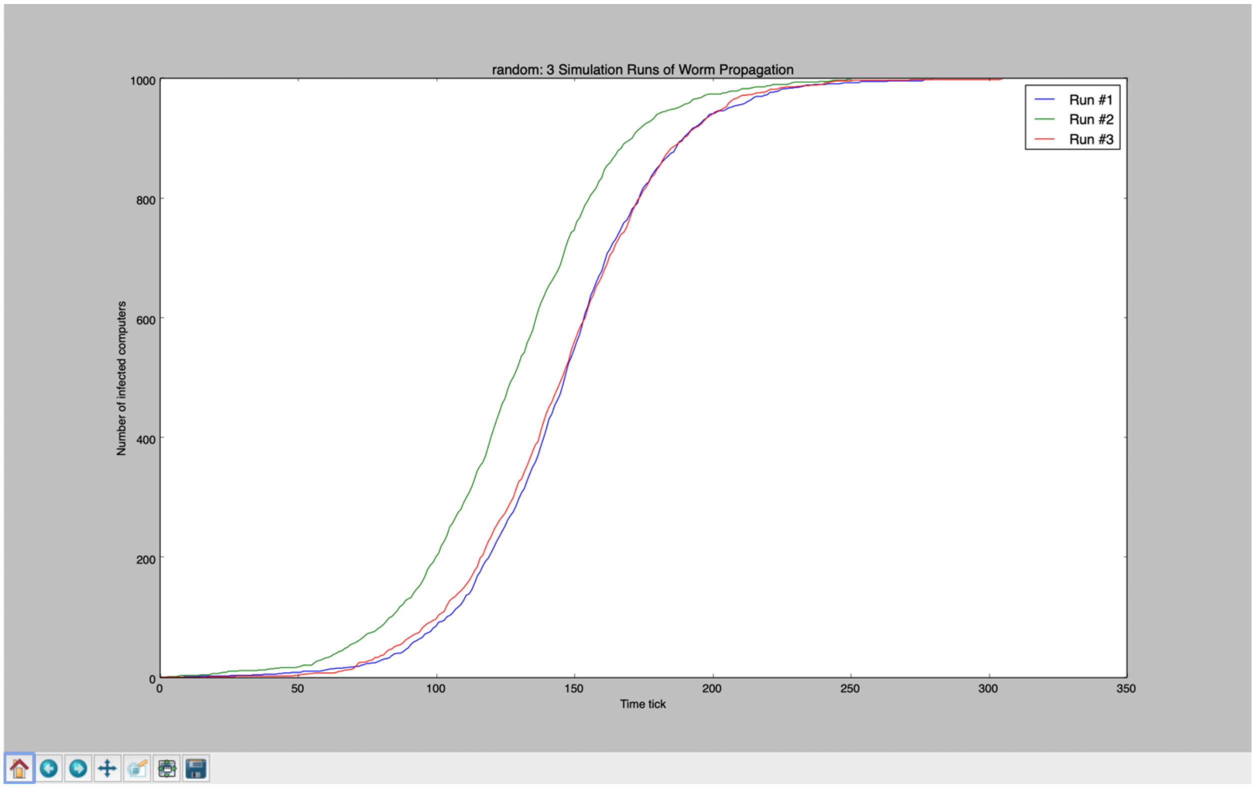
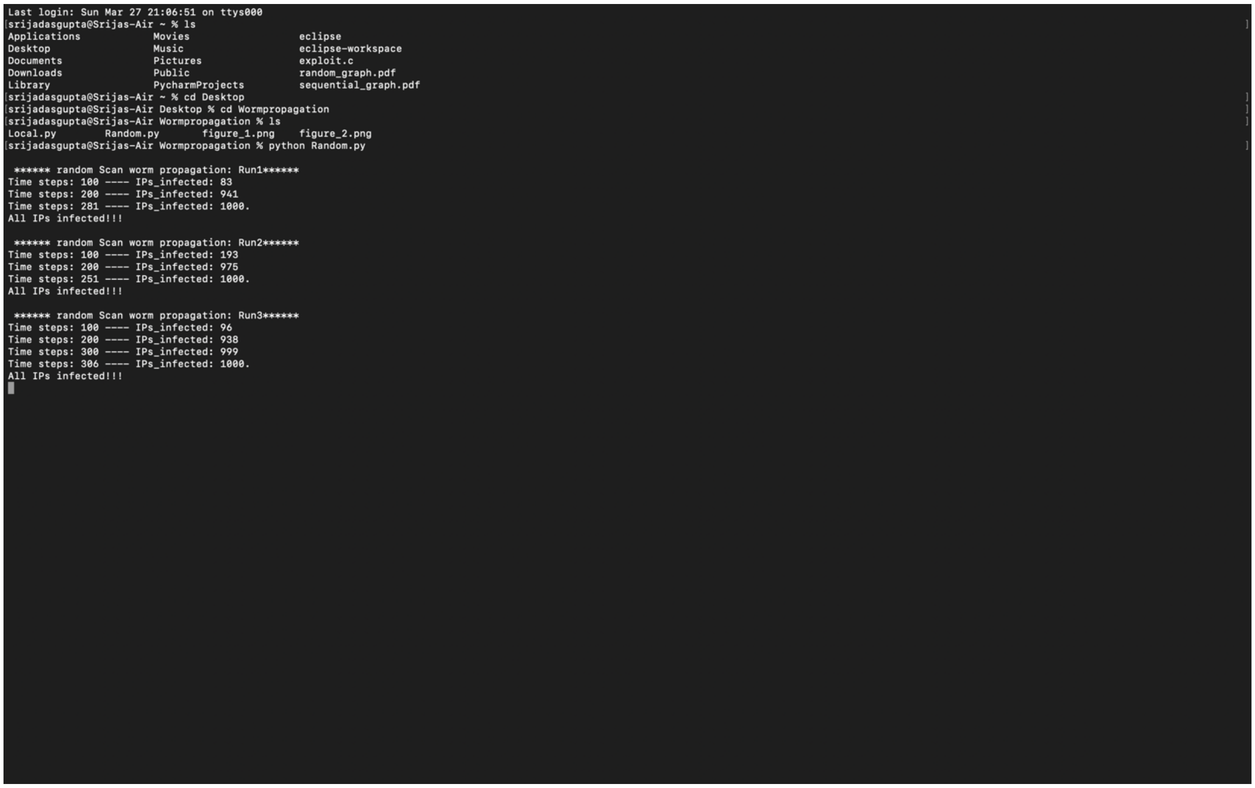


Figure 2: Three simulation run on worm propagation using sequential scanning method

Also, we observed the number of time steps required by each method to infect all of the 1000 vulnerable computers in the network which is shown in the table 1. From this data, we can conclude that sequential scanning spread worms faster in the network than the random scanning method.

Time steps to infect all 1000 computers



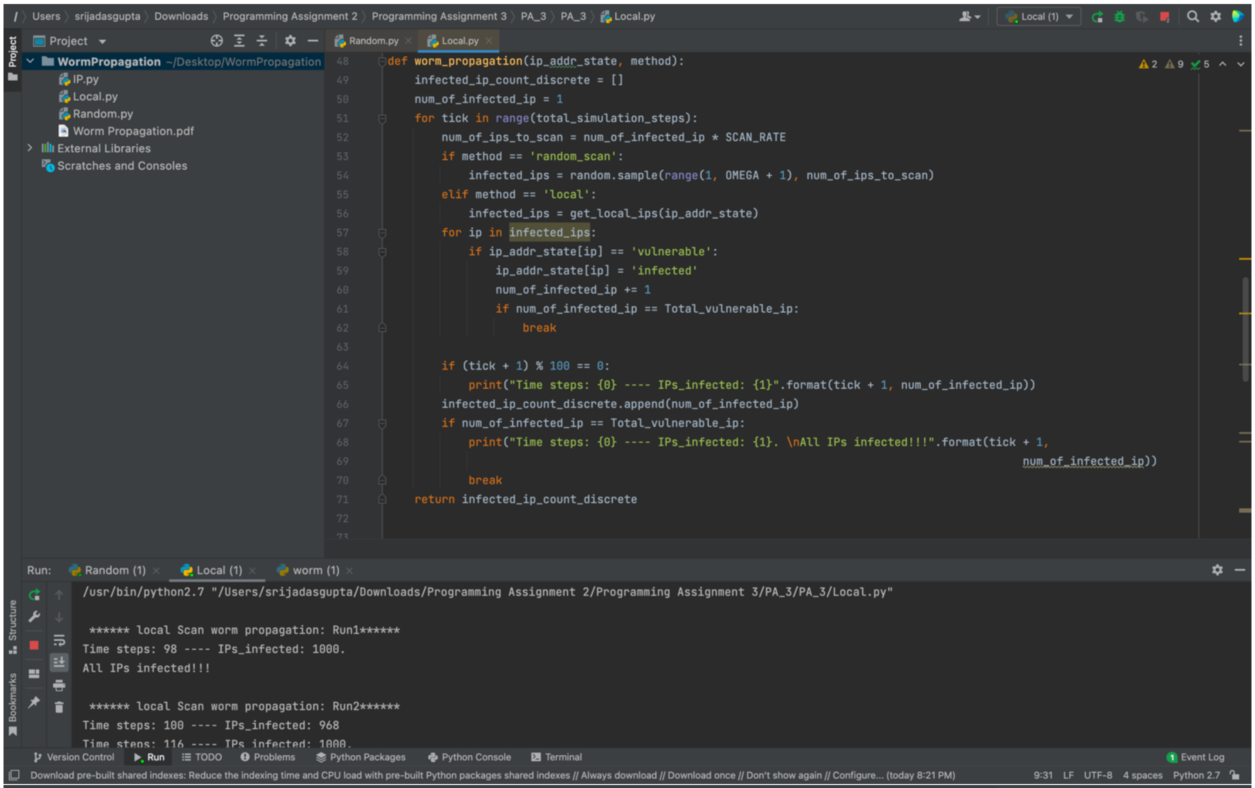
```
Last login: Sun Mar 27 21:06:51 on ttys000
srijadasgupta@Srijas-Air ~ % ls
Applications            Movies                  eclipse
Desktop                 Music                   eclipse-workspace
Documents               Pictures                exploit.c
Downloads               Public                  random_graph.pdf
Library                 PycharmProjects         sequential_graph.pdf
srijadasgupta@Srijas-Air ~ % cd Desktop
srijadasgupta@Srijas-Air Desktop % cd Wormpropagation
srijadasgupta@Srijas-Air Wormpropagation % ls
Local.py        Random.py       figure_1.png    figure_2.png
srijadasgupta@Srijas-Air Wormpropagation % python Random.py

 ****** random Scan worm propagation: Run1******
Time steps: 100 ---- IPs_infected: 83
Time steps: 200 ---- IPs_infected: 941
Time steps: 281 ---- IPs_infected: 1000.
All IPs infected!!!

 ****** random Scan worm propagation: Run2******
Time steps: 100 ---- IPs_infected: 193
Time steps: 200 ---- IPs_infected: 975
Time steps: 251 ---- IPs_infected: 1000.
All IPs infected!!!

 ****** random Scan worm propagation: Run3******
Time steps: 100 ---- IPs_infected: 96
Time steps: 200 ---- IPs_infected: 938
Time steps: 300 ---- IPs_infected: 999
Time steps: 306 ---- IPs_infected: 1000.
All IPs infected!!!
```

For Random Scanning

Table 1: Table to show the time steps required to infect all the vulnerable computers in the network.

Simulation	Time		
	RUN1	RUN2	RUN3
Random	281	251	306
Sequential	136	99	104

The screenshot of the output of the program is shown in the figure 3 and 4.



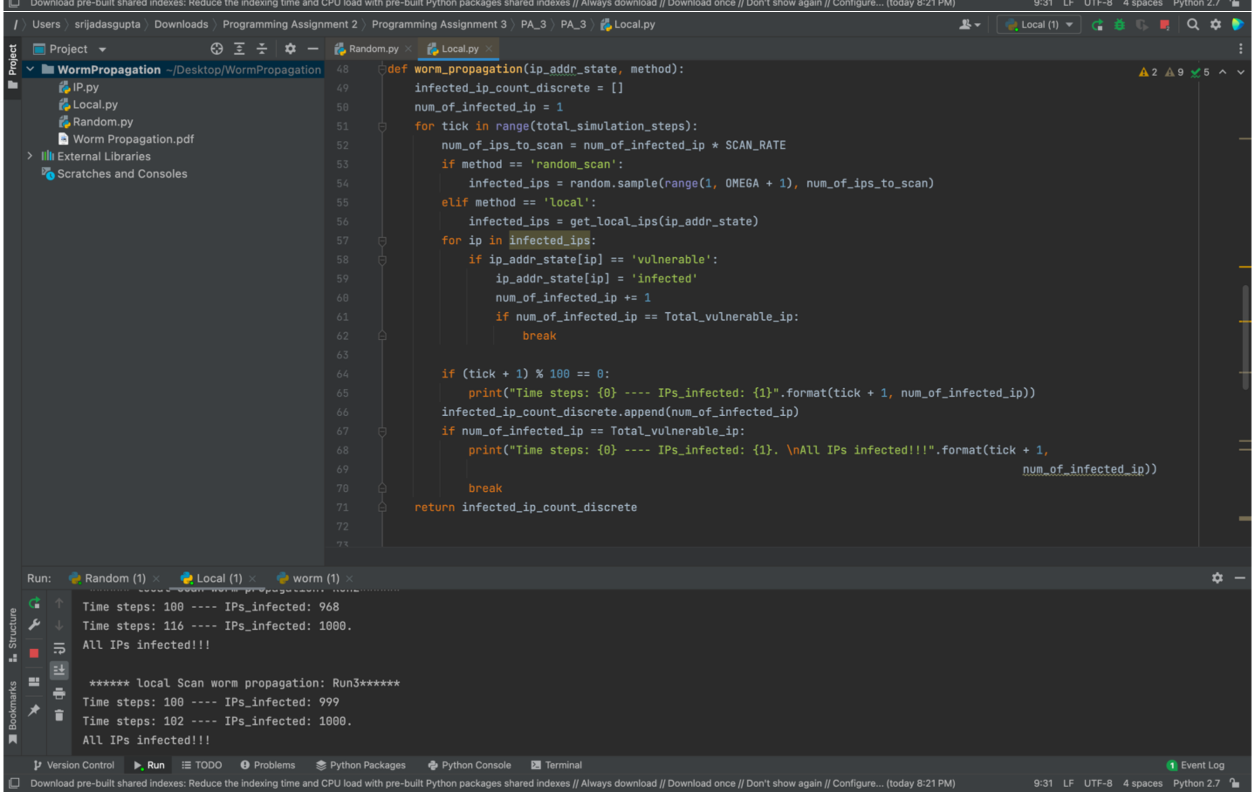


Figure 3: Output of Sequential scanning method

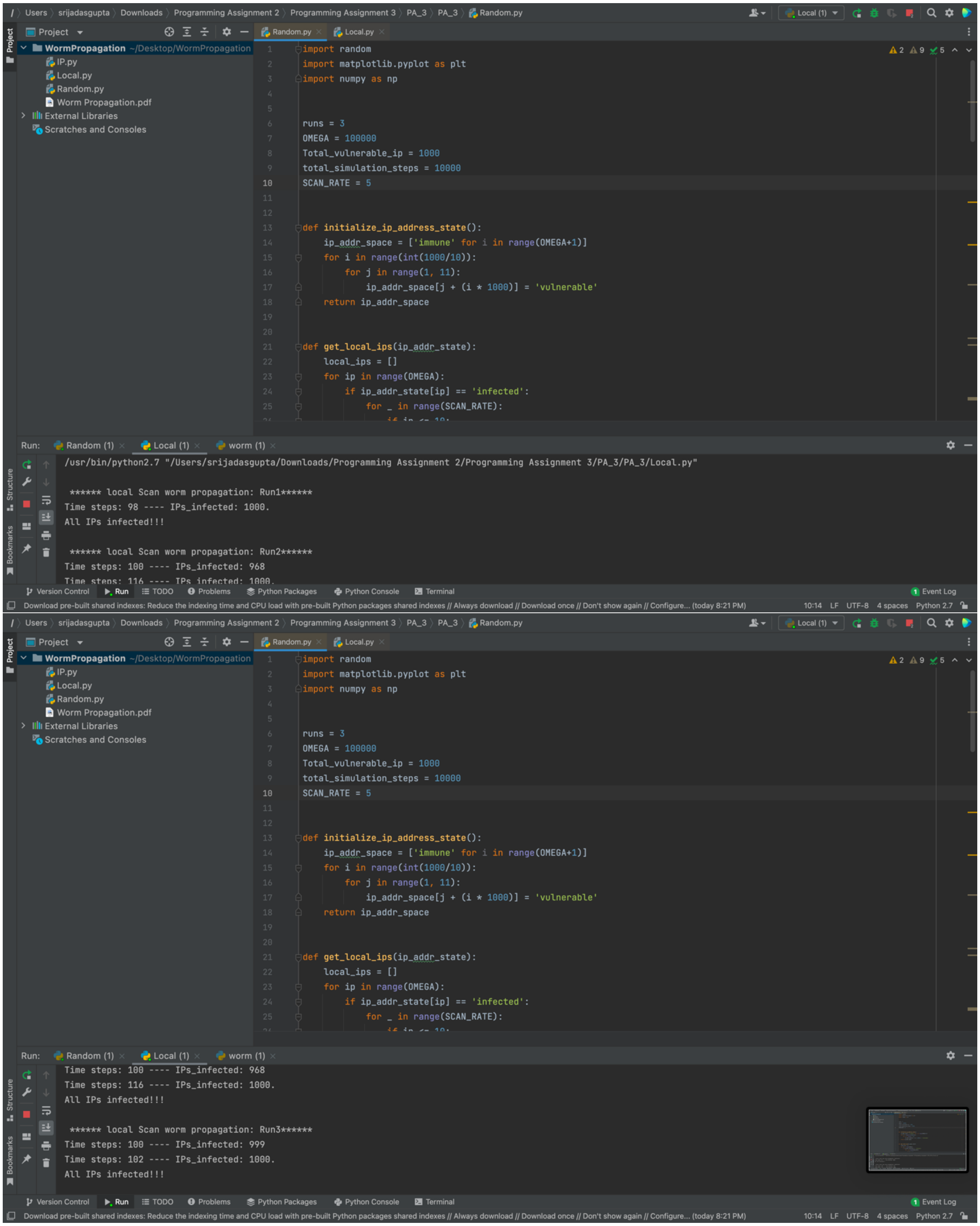
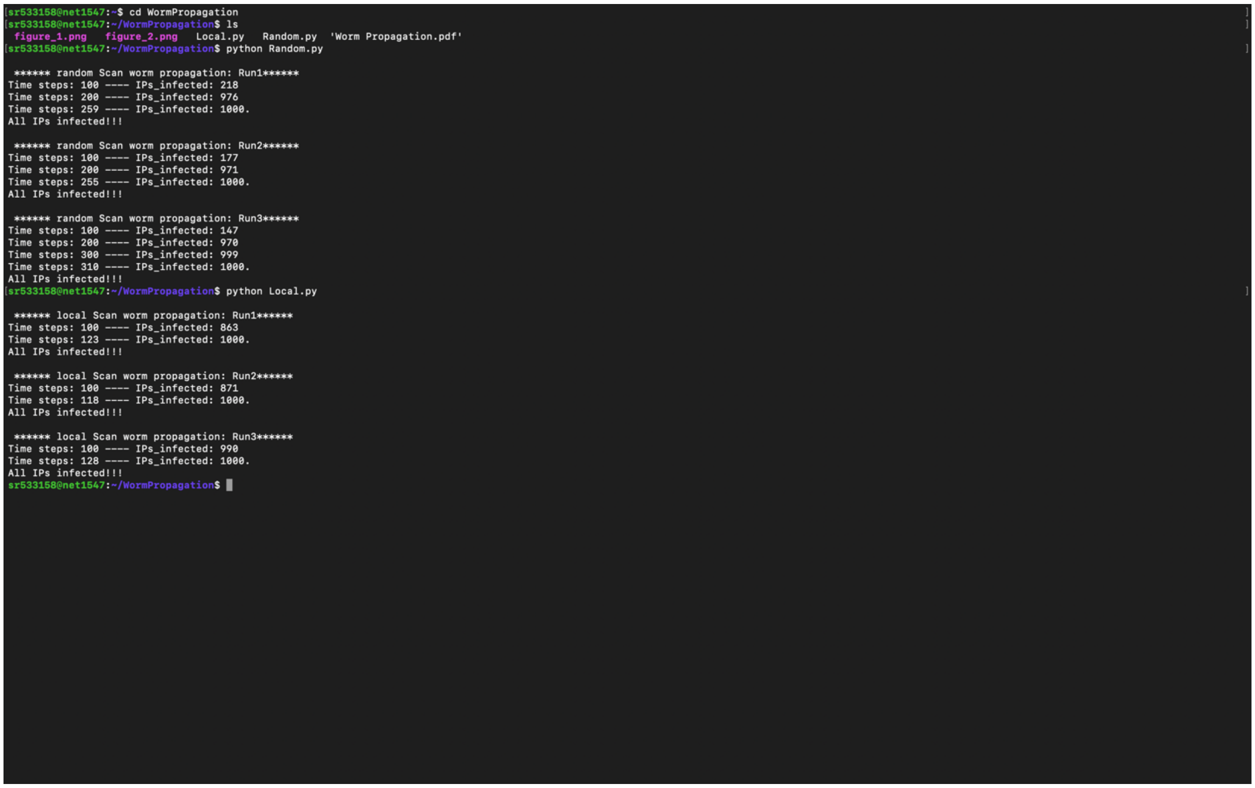


Figure 4: Output of random scanning method

5. Running in Eustis:



6.Instruction to run the code

I have submitted file called ”*WormPropagation.zip*” with this report. To run the code, follow the following instructions.

• Unzip ’WormPropagation.zip’ and change directory to ’WormPropagation’.

– $ unzip WormPropagation.zip

– $ cd WormPropagation

• Install prerequisite libraries.

– $ pip3 install numpy

– $ pip3 install matplotlib

• Run the code

– $ python3 random scan.py
– $ python3 sequential scan.py