Rongqi Sun (001537141)

Kang Shentu (001569432)

Program Structures & Algorithms

Spring 2021

Final Project – Team 19

1. Introduction

The spread of COVID-19 this year has severely affected all aspects of society, and it has also had a huge impact on our normal lives. In order to better understand the virus and contribute to the prevention of the virus, we hope to simulate the spread of this virus.

According to the data given by countries around the world, differences in policies adopted by countries of world ultimately led to different results of virus transmission, so it is necessary to explore the impact of different policies on virus transmission.

We complete this project by implementing a Python program. The program simulates real people' s movement and virus characters (such as: period of incubation). By the way, we also add some policies: mask, vaccine and social distance.

2. Goals

Simulate the spread of COVID-19:

- Describe the worst-case growth of any algorithms you create. Additionally, any data structures you employ must explain any invariant(s).
- Explain where you are getting entropy from for the simulation.
- In particular, compare at least two viruses with different k/R factors: the most obvious one would be SARS-CoV (“SARS”).
- You must write (and successfully run) unit tests for all of your formulas, rules, you’ re your code should have at least 60% coverage by line.
- The various parameters that you require must be defined via a configuration file (e.g. config.ini) that is easy to update.

3. Theory explanation

R0, the basic reproductive number, is defined as the mean number of infections caused by an infected individual in a susceptible population. According to "Data Analysis and Application" which belongs to the Guangdong Provincial Public Health Research Institute in China, we find that the R0 of COVID-19 is 5,7.

K is the overdispersion parameter of a negative-binomial distribution. This is just like the 80/20 rule. According to Estimating the overdispersion in COVID-19 transmission using outbreak sizes outside China by Akira Endo, we know that the k number of COVID-19 is 0.1, which means

that 80% of secondary transmissions may have been caused by 10% of infectious individuals.

For simplifying calculation, we want to calculate the average rate of spread named ‘S’ . After reading some data that our government published, we think a person will contact 80 people in a period of time. So, S = 5.7/80.

Considering K factor, super spreader S1=(S*0.8)/0.1, Normal spreader S2=(S*0.2)/0.9.

4. Complexity, invariants, entropy source

Complexity analysis

Actually, we use every single shape to stand for one person, so the use of memory and CPU will be large. Meanwhile, we try our best to decrease complexity of our algorithms by most are O(n). I will list some exceptions.

```
while Person.infected_num < Person.total_num:

    title = "No. %d Day    All cases: %d    Death cases: %d    Total number of people: %d" % (
        day, Person.infected_num, Person.dead_num, Person.total_num)
    #print(title)
    turtle.title(title)

    for a in people:
        for b in people:
            if id(a) == id(b):
                continue
            if a.status == 0 and b.status > 0 and dis(a, b) < DANGER_DIS:
                if not a.vaccine and not b.super:
                    a.infect(INFECTED_RATE[a.mask][b.mask])
                elif a.vaccine and not b.super:
                    a.infect(INFECTED_RATE_WITH_VACCINE[a.mask][b.mask])
                elif not a.vaccine and b.super:
                    a.infect(INFECTED_RATE_SUPER[a.mask][b.mask])
                else:
                    a.infect(INFECTED_RATE_WITH_VACCINE_SUPER[a.mask][b.mask])
    for a in people:
        a.move()
    turtle.update()
    #      time.sleep(1 / 300)
    count += 1
    # count==100, add one day
    if count > 100:
        print(title)
        day += 1
        count = 0
        for p in people[:]:
            if p.day() == -1:
                p.dead()
                people.remove(p)
```

This is the most complex one. O will close to infinity in worst case but there is no doubt that finally everyone will be infected according to theory. The loop will terminate.

```python
for a in people:
    for b in people:
        if id(a) == id(b):
            continue
        if a.status == 0 and b.status > 0 and dis(a, b) < DANGER_DIS:
            if not a.vaccine and not b.super:
                a.infect(INFECTED_RATE[a.mask][b.mask])
            elif a.vaccine and not b.super:
                a.infect(INFECTED_RATE_WITH_VACCINE[a.mask][b.mask])
            elif not a.vaccine and b.super:
                a.infect(INFECTED_RATE_SUPER[a.mask][b.mask])
            else:
                a.infect(INFECTED_RATE_WITH_VACCINE_SUPER[a.mask][b.mask])
```

This is O(N^2) in worst case.

Invariants

Almost every invariant is defined in 'config.py'.

```python
TOTAL_W = 800  # area's width
TOTAL_H = 500  # area's height
DANGER_DIS = 35  # when distance closer than this, may be infected

# Infection Rate -COVID-19
#
COV_R0 = 5.7
COV_K = 0.1
COV_S = COV_R0 / 80  # 7.125%
COV_S1 = COV_S * 0.8 / COV_K #super spreader (s*0.8)/k
COV_S2 = COV_S * 0.2 / (1 - COV_K) #normal spreader (s*0.2)/(1-k)
COV_EV = 0.9  # mRNA vaccine effectiveness

# Diagnose Rate
Diagnose_Rate = 0.9
# Death Rate
DEATH_Rate = 0.05
# People amount settings
TOTAL_NUM = 500
INFECTED_NUM = 1
# random.randint(0, total_num)
HEALTHY_NUM = TOTAL_NUM - INFECTED_NUM
MASKED_RATE = 0.5  # rate of people wear masks

VIRUS_LATENCY = 7

IR_WW = 0.9  # The infection rate of both without a mask
IR_MM = 0.01  # both with a mask
IR_WM = 0.5  # the infected without a mask, the healthy with
IR_MW = 0.3  # the infected with a mask, the healthy without
# rate of becoming super spreader
SUPER_RATE = COV_K
```

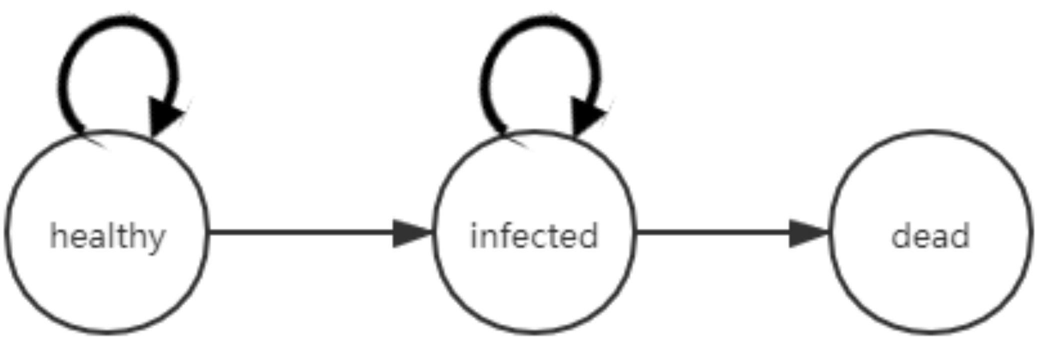
```
#normal spreader
INFECTED_RATE = [[IR_WW * COV_S2, IR_MW * COV_S2], [IR_WM * COV_S2, IR_MM * COV_S2]]
# the healthy with vaccine
INFECTED_RATE_WITH_VACCINE = [[IR_WW * COV_S2 * (1 - COV_EV), IR_MW * COV_S2 * (1 - COV_EV)],
                              [IR_WM * COV_S2 * (1 - COV_EV), IR_MM * COV_S2 * (1 - COV_EV)]]

VACCINE_RATE = 0.4  # rate of people get vaccine
# infected rate of super spreader
INFECTED_RATE_SUPER = [[IR_WW * COV_S1, IR_MW * COV_S1], [IR_WM * COV_S1, IR_MM * COV_S1]]
# the healthy with vaccine
INFECTED_RATE_WITH_VACCINE_SUPER = [[IR_WW * COV_S1 * (1 - COV_EV), IR_MW * COV_S1 * (1 - COV_EV)],
                              [IR_WM * COV_S1 * (1 - COV_EV), IR_MM * COV_S1 * (1 - COV_EV)]]



# Infection Rate -SARS
# SARS_R0 = 3
# SARS_S = SARS_R0/80 #3.75%
# IR_WW = 0.9  # The infection rate of both without a mask
# IR_MM = 0.01  # both with a mask
# IR_WM = 0.5  # the infected without a mask, the healthy with
# IR_MW = 0.3  # the infected with a mask, the healthy without
# #
# INFECTED_RATE = [[IR_WW*SARS_S, IR_MW*SARS_S], [IR_WM*SARS_S, IR_MM*SARS_S]]
# # no vaccine
# INFECTED_RATE_WITH_VACCINE = INFECTED_RATE
# VACCINE_RATE = 0
```

Entropy source

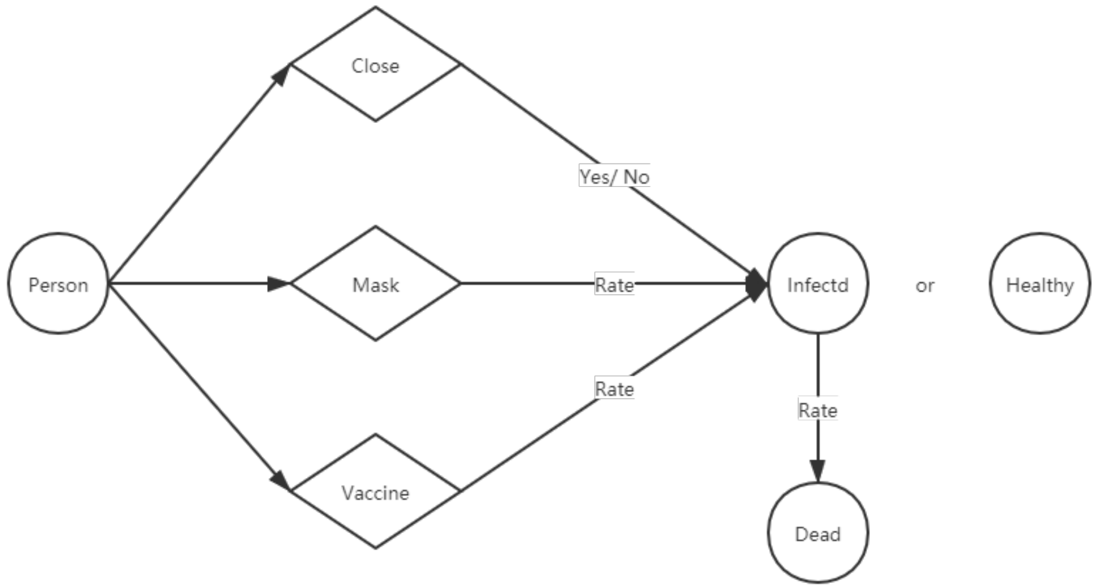


The number of people: N

In the beginning: N-1

h = lg(N!) ~ NlgN = 500lg500 = 4500

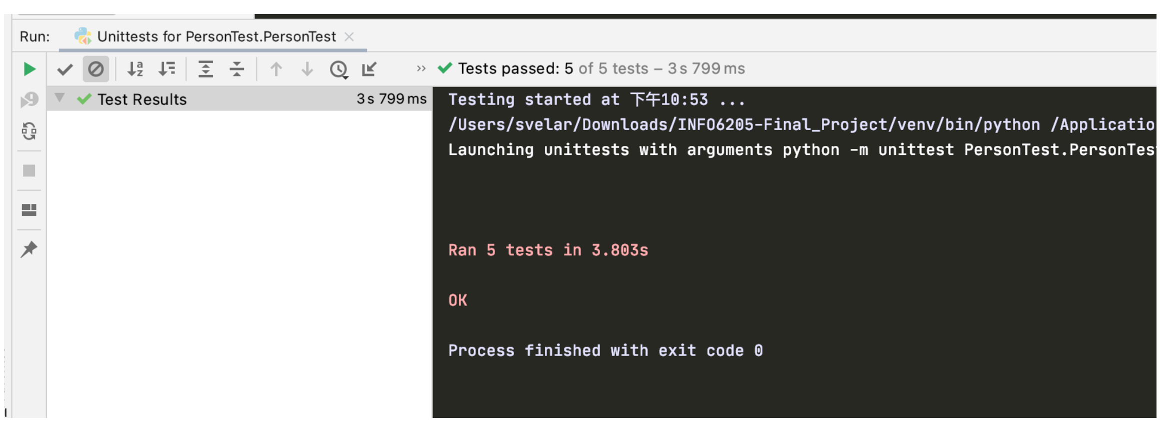
5. Flow



P.s: close means closer than social distance

Detailed implementations are all on repo and you will understand by comments.

6. Unit test



7. Output and conclusion

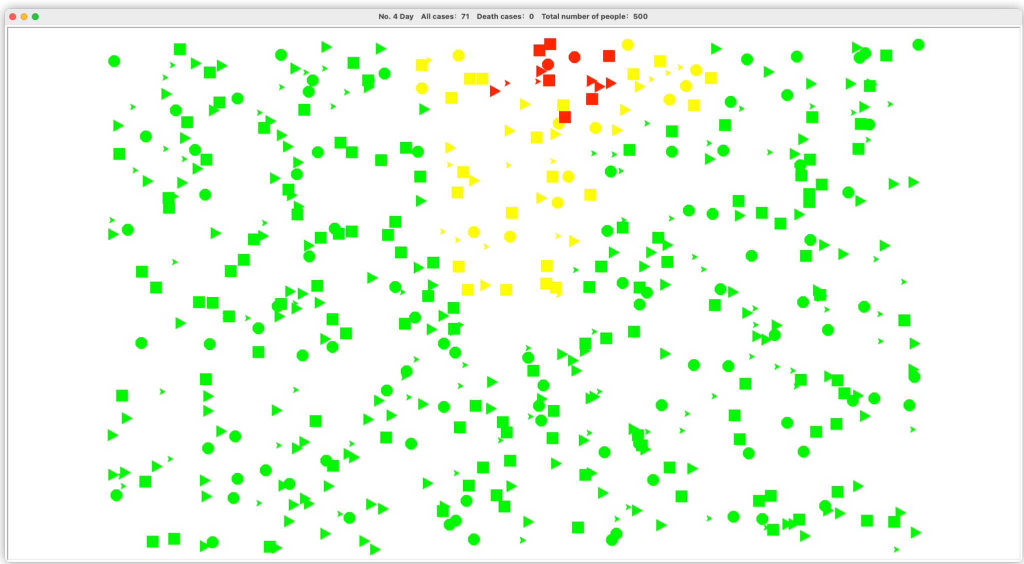
Of course, changing rate of wearing masks and injecting vaccines will affect terminated time.

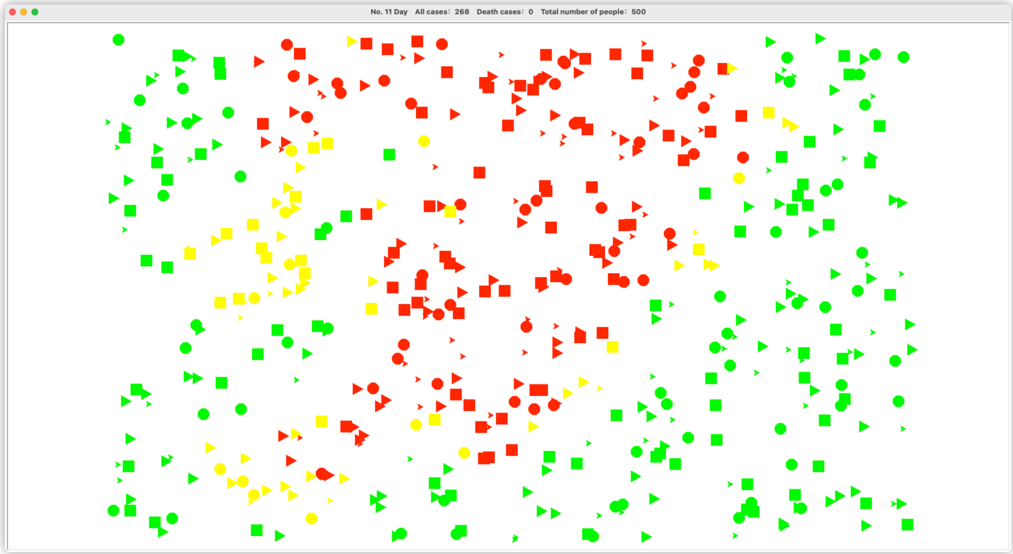
Below are with rate of wearing masks is 0.5 and rate of vaccine is 0.4. By the way, I uploaded video with this progress.

By the way, I uploaded video with this progress.

Links:

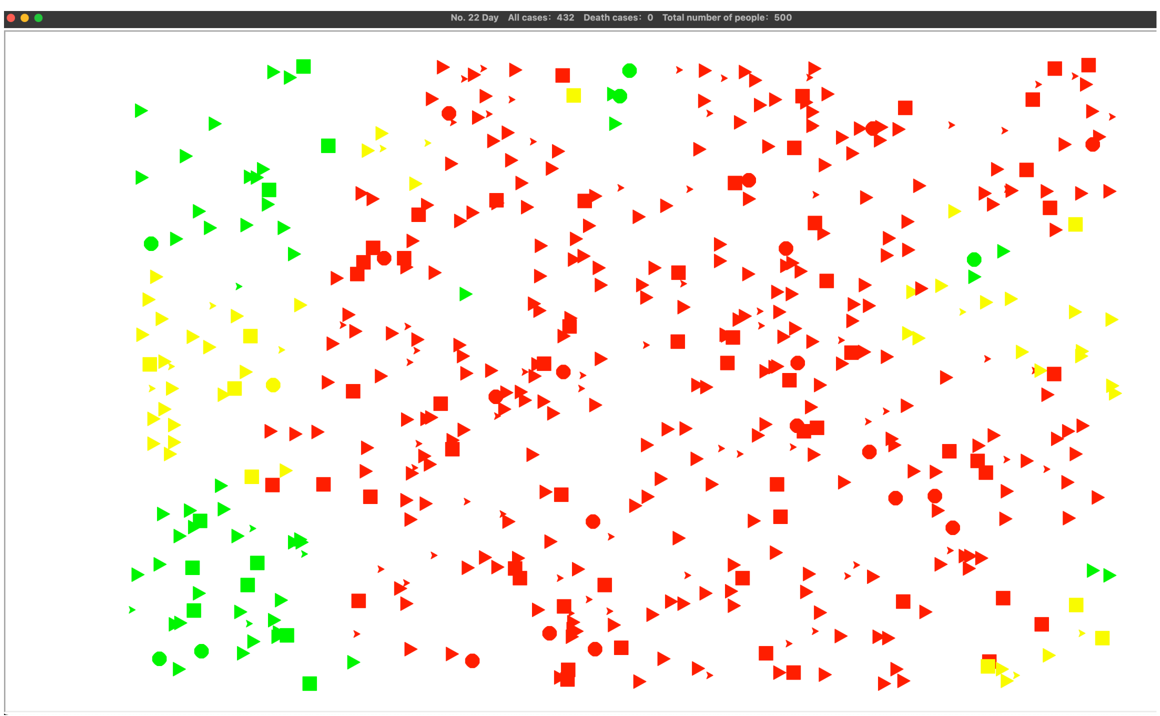
https://drive.google.com/file/d/1VjEgpxDlHPlnxXvTKq042f-5DPby8RYO/view?usp=sharing






```
No. 40 Day     All cases: 499     Death cases: 0     Total number of people: 500
('Time of all infected', datetime.timedelta(0, 565, 257034), 'Days:', 41)
```

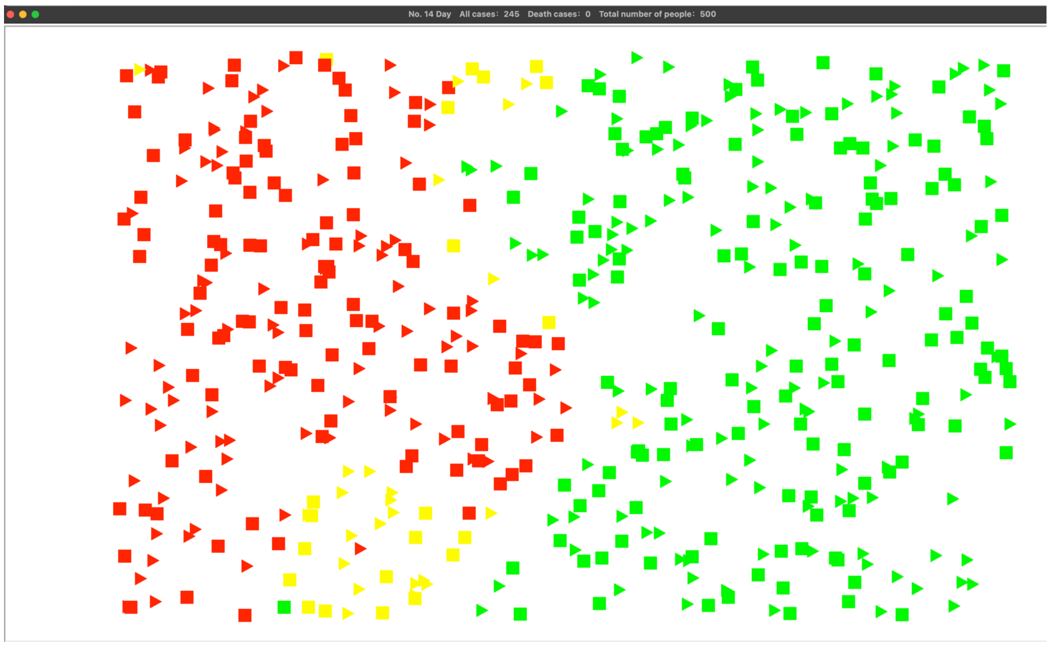
Below are with rate of wearing masks is 0.3 and rate of vaccine is 0.3.




```
No. 35 Day     All cases: 499     Death cases: 0     Total number of people: 500
('Time of all infected', datetime.timedelta(0, 502, 57456), 'Days:', 36)
```

And for SARS. The only thing we need to do is to change config. Because it has no vaccine, rate of vaccine is 0. And change R0=3.0, S and k.

Below are SARS with rate of wearing masks is 0.5 and rate of vaccine is 0.




```
No. 56 Day    All cases: 499    Death cases: 0    Total number of people: 500
No. 57 Day    All cases: 499    Death cases: 0    Total number of people: 500
('Time of all infected', datetime.timedelta(0, 881, 825473), 'Days:', 58)
```

Conclusion:

1. When we do nothing to the simulation, it will spread quickly.

2. Some policies such as wearing masks, getting vaccines and keeping social distance will stop the spread of pandemic in some way.

3. Reducing mobility is an effective way to slow the spread. So keep working from home.

4. There are some areas that virus spreads quicker than others because of super spreaders, or the k factor.

5. The simulation of SARS compared with COVID-19, we can find as R0 decreases, it spread a lot slower. But in fact, it has higher rate of death.

6. I am sure that virus will terminate one day.

8. Reference

https://www.sciencefocus.com/the-human-body/what-is-the-r-number-and-why-is-itrelevant-to-coronavirus/

https://www.zhihu.com/question/371079927

https://www.worldometers.info/coronavirus/

https://en.wikipedia.org/wiki/Coronavirus_disease_2019

https://www.sohu.com/a/387326620_120059072