Breadth First Search in Tracking the Spread of an Epidemic Disease

Byan Sakura Kireyna Aji 13518066
Informatics Engineering
School of Electrical Engineering and Informatics, ITB
Bandung, Indonesia
E-mail: byansakura14@gmail.com

***Abstract*—The hysteria of a new outbreak often happens late and occurs only when significant number of deaths has paid the cost as a result of a mass incomprehension in how such disease spread. Such problem can be solved using a simple Breadth First Search algorithm combined with graph and some basic calculus by showing how the travel rate and the number of populations in a certain area may contribute to the spread of a disease.**

Keywords—breadth first search, graph, virus

I. Introduction

In every decade it is not scarce for a new pandemic to surface and cause wide infection that may affect the life of a certain population. In some case the outbreak may come insignificant, circling only a certain area and can be taken under control withing a short time. The rest of the case sometime is harder to handle, especially when it happens worldwide and causing multination mass with rapid spreading. Before one knows it, a worldwide quarantine turns into a mandatory and every system that has a connection with economy, which in nowadays case it is all system, is failing.

It is said that such situation is out of human's scope of hand, something that is unavoidable and bound to happen in a way or another. What if such statement is wrong? Everything can be prevented in the right way and this is where computer science may come to save the day. Maybe not now, but no one knows what may happen in near future.

Searching algorithm has been developed throughout the years that it now has a lot of branch that can be applicated in a lot of cases that is close to this one. The nearest approach is the breadth first search algorithm. This algorithm shows branching in an order of a queue by traversing the graph from the root node and explores all the neighboring nodes. The exploration starts from the nearest node to all unexplored nodes until the goal state is satisfied.

The combination of this searching algorithm with a function to decide whether an area is already infected or not can show a map of the infected areas from day to day. This information may seem trivial but it can be a good and valid prediction of what the future may hold so that the authority can take action in order to slow down the rate of the spread.

II. Theory

A. Graph

Before jumping into how the Breadth First Search algorithm works, one must first understand what graph is. Graph is a non-linear data structure that consists of nodes and edges where the nodes are often referred to as vertices while the edges are the arcs that link any two nodes within the graph.

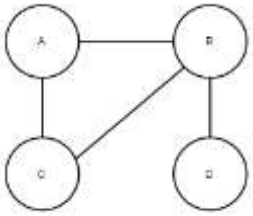

Figure 1. Example of Graph

Graph is often used to solve a lot of problem by representing itself as network that can be translated as circuit, cable, or sometime like the case we are currently studying, map of an area. By that, each node is a structure and contains information according to what the graph may represent.

There are a lot of graph varieties that differ on their structures of nodes and edges. Some graph is pictured with arrow to show the direction. This kind of graph is called the directed graph.

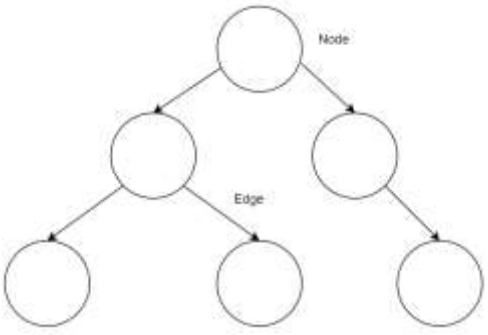

Figure 2. Example of Directed Graph

B. *Breadth First Search*

Breadth first search is an algorithm for traversing graph data structures that start from the arbitrary node of a graph and explores all the neighbor nodes at the present depth prior to moving on to the nodes at the next depth level.

This algorithm implements the queue principle of *first in first out* where this check whether a vertex has been discovered before enqueueing the vertex. Each child node would be explored thoroughly before moving on to the child of the child.

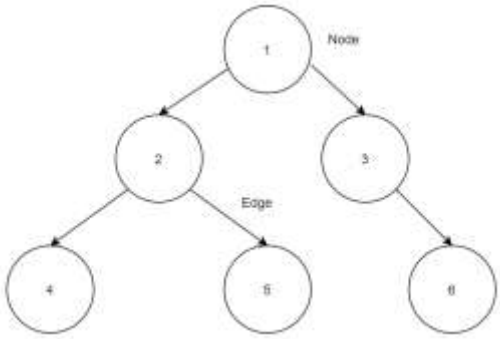

Figure 3. The order of search in breadth first search

As can be seen in "Fig. 3", the order of exploration goes from the parent node to all of its children before the children of a child is explored.

TABLE I. THE ITERATION OF EXPLORATION

Iteration	V	Q	Visited					
			1	2	3	4	5	6
Initialize	1	{1}	T	F	F	F	F	F
1	1	{2,3}	T	T	T	F	F	F
2	2	{3,4,5}	T	T	T	T	T	F
3	3	{4,5,6}	T	T	T	T	T	T
4	4	{5,6}	T	T	T	T	T	T
5	5	{6}	T	T	T	T	T	T
6	6	{}	T	T	T	T	T	T

The initialization starts in the node 1, once it has been explored, the node 2 and node 3 are added to the queue. Next in line for the checking is the node 2, adding node 4 and node 5 to the queue behind 3. Next node to check is back to 3 where later the node 6 as its child is added to the queue behind 5.

The execution goes on in the order of queue with 4 as the preceding node, followed by node 5 and ended by node 6.

Meanwhile, the algorithm of the breadth first search is best expressed as the following

```
procedure BFS(G, start_v) is
    let Q be a queue
    label start_v as discovered
    Q.enqueue(start_v)
    while Q is not empty do
        v := Q.dequeue()
        if v is the goal then
            return v
        for all edges from v to w in G.adjacentEdges(v)
            if w is not labeled as discovered then
                label w as discovered
                w.parent := v
                Q.enqueue(w)
```

The time and complexity of this algorithm can be expressed as O(|V|+|E|) since every vertex and every edge will be explored in the worst case. |V| represent the number of vertices while |E| symbolizes the number of edges in the graph.

Even though this algorithm is quite space consuming, it is guaranteed that this algorithm will find the goal state if one exists or in other word, this algorithm is complete.

III. APPLICATION

By representing map of cities as nodes and their connection with each other as edges in graph, one can implement the breadth first search algorithm to run one by one examination of the cities within the queue. This algorithm will create a queue insisting a collection of cities that have to be visited according to the closure with the other cities.

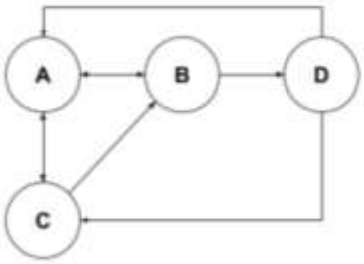

Figure 4. The potray of cities as nodes

In doing so, some information is necessary to obtain the function that decides the spread of the disease from a city to another. The information includes the source of the disease as in where it first surfaces to know the search key of the node, the population that inhibit each city, the time when the disease infects a city and the total of day since the first infection happened.

With the help of constant variable, the declaration of the information above can be put into function with population as P(A), time when the disease appears as T(A), and the total of day as t(A) with A as the name of the city.

The population of the infected can be defined by the function I(A,t(A)) that is independent for each city.

$$I(A,t(A)) = \frac{P(A)}{1+(P(A)-1)e^{-\gamma t(A)}}, \gamma = 0.25 \quad (1)$$

The chance of a person from a City A to travel to City B is defined by the function Tr(A,B) with B as the destination city.

The spread of the disease from the City A to the City B is constant and formulized as the function S(A,B) where if the outcome of the function S is bigger than 1 then the destination city is put under infection.

$$S(A,B) = I(A,t(A)) \times Tr(A,B) \quad (2)$$

A. Analysis

The breadth first search algorithm can take place in this code script procedure as a query that request an input as the source city and the targeted day. This procedure will do a reset to all cities and start the first day of infected day to 0 and make a new queue of cities.

As the queue is not empty, this procedure will update the infected cities by running a check of whether transmission of disease take place in that city. This procedure will be repeated until all cities are thoroughly checked. The checking will continue with the next day as the beginning day.

The infected city will be depicted using the red node in contrast with the uninfected city that will be depicted in white node. The arrow of the spread will also turn dark red if in the arrow portrays the transmission from one city to another.

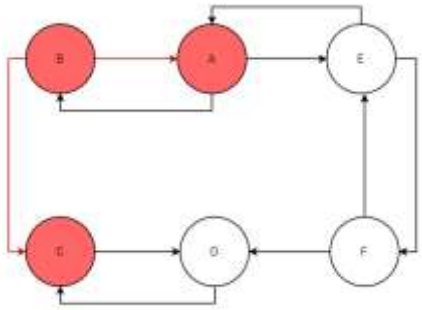

Figure 5. The depictation of the city's state

The spread is checked by the function S with the starting city and the destination city as the parameter that return the result value of the multiplication of the infected population in a certain day with the possibility that someone may travel from a starting city to the destination city. A city is confirmed if the result of the function S described in (2) is bigger than 1.

Since the calculation of the transmission has to know the length of days one city suffered from infection since it is infected, not since the first disease appears, there is another need to calculate when exactly that city is infected with the disease by running a check on (2).

$$S = \frac{P(A).Tr(A,B)}{1+(P(A)-1)e^{-\frac{t^1}{4}}} > 1$$

$$P(A).Tr(A,B) > 1+(P(A)-1)e^{-\frac{t}{4}}$$

$$e^{-\frac{t}{4}} < \frac{P(A).Tr(A,B)-1}{(P(A)-1)}$$

$$-\frac{t}{4} < ln(\frac{P(A).Tr(A,B)-1}{(P(A)-1)})$$

$$t > -4ln(\frac{P(A).Tr(A,B)-1}{(P(A)-1)}) \quad (3)$$

The breadth first search algorithm will later check if each city goes through the path between cities so that the complexity of this algorithm can be declared as O(n) with n as the number of cities.

B. Illustration

To draw a better picture, here is an example of the population configuration as a study case.

4 Cities, Source: WHN
WHN Population: 4000
BDG Population: 2000
MLY Population: 3000
HKG Population: 1000

There are four cities in the map with the WHN as the source of disease. The population of each city is described as above with 4000 in WHN, 2000 in BDG, 3000 in MLY, and 1000 in HKG.

And this is the example of the map configuration.

8 Paths
Tr(BDG, WHN) = 0.02
Tr(BDG, MLY) = 0.0015
Tr(WHN, BDG) = 0.05
Tr(WHN, HKG) = 0.0007
Tr(MLY, BDG) = 0.003
Tr(MLY, WHN) = 0.004
Tr(HKG, BDG) = 0.002
Tr(HKG, MLY) = 0.001

Accordingly, there are 8 paths connecting the cities with the probability that someone travels from one city to another depicted in the Tr function of the two cities in the example above.

If T=20 in this case, the initialization of the queue Q can be defined as [WHN→BDG, WHN→HKG]. WHN will be portrayed in a red node as it is the source of the infection in day 0.

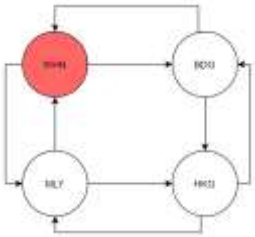

Figure 6. The initial state

The checking:

- WHN->BDG

The disease first appears in Wuhan, so that
T(WHN)=0
t(WHN) = 20-0.
The amount of infected person in WHN:
I(WHN,20) = 143,2.
The probability of transmission:
S(WHN,BDG) = I(WHN,20)x Tr(WHN,BDG)
S(WHN,BDG) = 143,2 x 0.05
S(WHN,BDG) = 7
Since S(WHN,BDG)>1, the disease has been successfully transmitted to BDG.
Then find out the day when the disease gets into Bandung, which is t(BDG)=13.
So, the disease will spread to BDG 13 days after the outbreak.
T(BDG) = 13+T(WHN) = 13.
q=[WHN->HKG, BDG->MLY]

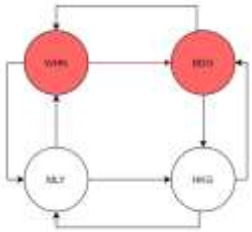

Figure 7. BDG's changing state

- WHN->HKG

The disease first appears in Wuhan, so that
T(WHN)=0
t(WHN) = 20-0.
The amount of infected person in WHN:
I(WHN,20) = 143,2.
The probability of transmission:
S(WHN,HKG) = I(WHN,20) x Tr(WHN,HKG)
S(WHN,HKG) = 143,2 x 0.0007
S(WHN,HKG) = 0.1
Since S(WHN,HKG)<1, the virus don't spread to HKG. The queue Q is now
q=[BDG->MLY]

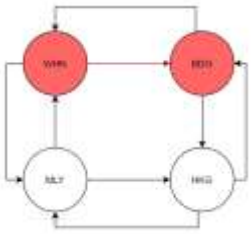

Figure 8. The current state

- BDG->MLY

The disease spreads from BDG in 13'rd day
t(BDG) = 13.
The amount of infected person in BDG:
I(BDG,20) = 138,2.
The probability if transmission:
S(BDG,MLY) = I(BDG,20) x Tr(BDG,MLY)
S(BDG,MLY) = 143,2 . 0.0015
S(BDG,MLY) = 0.2.
Since S(BDG,MLY)<1, the virus don't spread to HKG. The queue Q is now

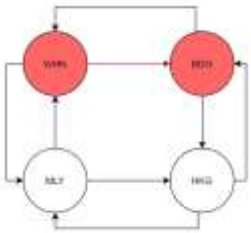

Figure 9. The current state.

From the calculation that is shown in the graphs above, it can be concluded that within 20 days, the transmission may happen to BDG from WHN.

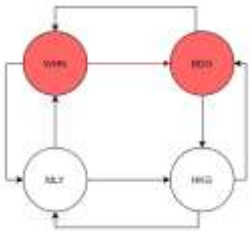

Figure 10. The final state.

C. Implementation

This combined breadth first search algorithm and the deciding function can be put into an application that require the information of the cities as the input so it can show the probability of the disease's movement in the questioned day.

In an easier model, this application is going to show a colored graph for output as a reference to the actual map. The graph can show the change of the spread day by day by changing some queries.

Figure 11. The preview of the application

As shown in fig. 8, the application requests some input of the current information. The first line asks for the day after the first outbreak. The graph column requires user to fill it with the number of cities and the source city, then all the name of the cities followed by its population. The connectivity map has to be filled with the number of paths between the cities and the probability of someone traveling from one city to another using each path.

```
4 A
A 1000
B 5000
C 1000
D 1000
```

```
8
AB 0.2
AC 0.05
BA 0.5
BD 0.75
CA 0.1
CB 0.1
DA 0.5
DC 0.1
```

Figure 12. The inputs for day 0

The output for the day 0 is going to be one red node which is the starting city itself because no infection may transmit for the time being.

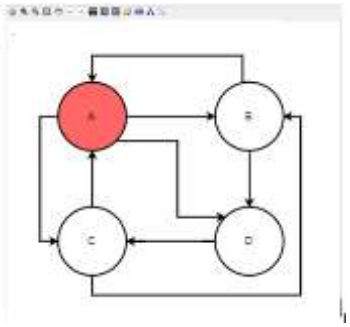

Figure 13. The result of day 0

To see what happens in the next day to come, a change can be made in the day column as shown in fig. 11 below.

Figure 14. The inputs for day 10

Meanwhile the tenth day has shown a change in both the color of the nodes with B and D turning red and the paths between A to B, B to A, and B to D turning dark red.

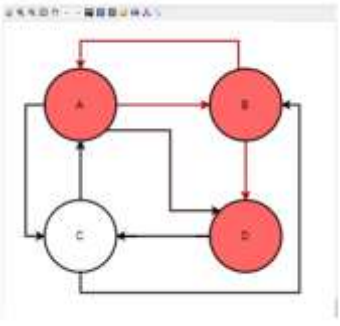

Figure 15. The result of day 10

IV. Conclusion

Prevention is always the best action taken in most of outgoing problem. The bigger spread can always be contained in the right way and thankful to the development of computer science nowadays, we can predict what might happen in the future with the correct information given.

This idea of application is more than just diagrams of prediction as it can always be developed further to show some more feature like maximum traveling ratio from one city to another to suppress the infection. This can be applied in the real world during social distancing in putting a limitation in the number of people going in or out the city.

Indeed, there is more than just breadth first search algorithm for traversing graphs like the case we are studying. However, it is safe to be concluded that the breadth first search is more than just the closest approach, but also the best approach as the checking in this algorithm is run thoroughly to all children of a node in the queue so that if a spread happen in one child, another spread would first occur in its sibling rather than its child.

Video Link at Youtube

https://youtu.be/nQUJwg7Fd88

Acknowledgment

I would like to express my gratitude toward Mr. Rinaldi Munir for only because of his lecture that I am able to put one of his tasks in words of an essay. I also would love to thank my friends Nafkhan and Arthur for helping me turning this idea into an actual application.

References

[1] Munir, Rinaldi. 2018. *BFS dan DFS (2018)*. Bandung, Sekolah Tinggi Elektro dan Informatika Institut Teknologi Bandung. https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2019-2020/stima19-20.htm

[2] Garg, Prateek. 2016. *Breadth First Search*. source : https://www.hackerearth.com/practice/algorithms/graphs/breadth- first-search/tutorial/

STATEMENT

With this, I hereby state that this paper is my own writing, not a translation and not a plagiarization of someone else.

Bandung, 29 April 2020

Byan Sakura
13518066