BI-GY 7453: Prof. Manpreet S Katari
February 16, 2022

ASSIGNMENT 2 : Due Date : 25-February-2022

1. Criteria for evaluation

- Commenting of code
- Code file you have submitted needs to be executable and satisfy all the test cases
- Clear and comprehensive explanation of your problem solving approach

Question 1. Given an adjacency list representation of a Graph write a program to convert it into Adjacency Matrix representation?

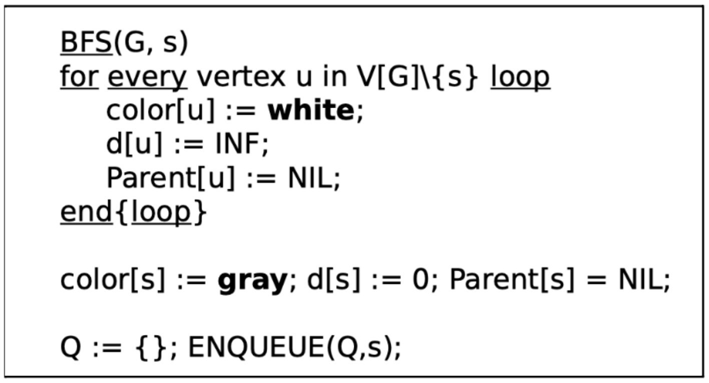
Input. Adjacency List:
$0 \rightarrow 1, 4$
$1 \rightarrow 0, 2, 3, 4$
$2 \rightarrow 1, 3$
$3 \rightarrow 1, 2, 4$
$4 \rightarrow 0, 1, 3$

Output. Adjacency Matrix:

	0	1	2	3	4
0	0	1	0	0	1
1	1	0	1	1	1
2	0	1	0	1	0
3	0	1	1	0	1
4	1	1	0	1	0

Question 2. This question has three parts 2.1,2.2 and 2.3.
For this homework, we will use a gene network graph from Biogrid (https://thebiogrid.org/) for the species Arabidopsis thaliana. The gene interaction network is provided called large.graph. The Gene names in this species start with AT, followed by chromosome number, followed by G for gene, finally and unique identifier. For example, we will use AT5G49450 as our source.

2.1. Implement a breadth-first search algorithm using any programming language of your choice.

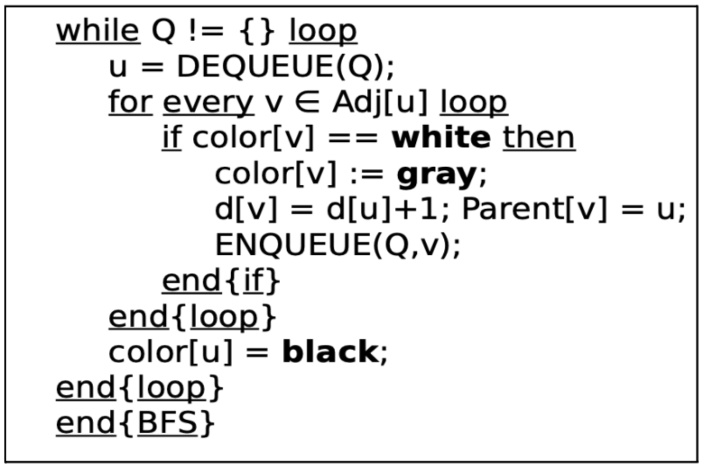


```
BFS(G, s)
for every vertex u in V[G]\{s} loop
    color[u] := white;
    d[u] := INF;
    Parent[u] := NIL;
end{loop}

color[s] := gray; d[s] := 0; Parent[s] = NIL;

Q := {}; ENQUEUE(Q,s);
```

Step 1: Initialize the vertices

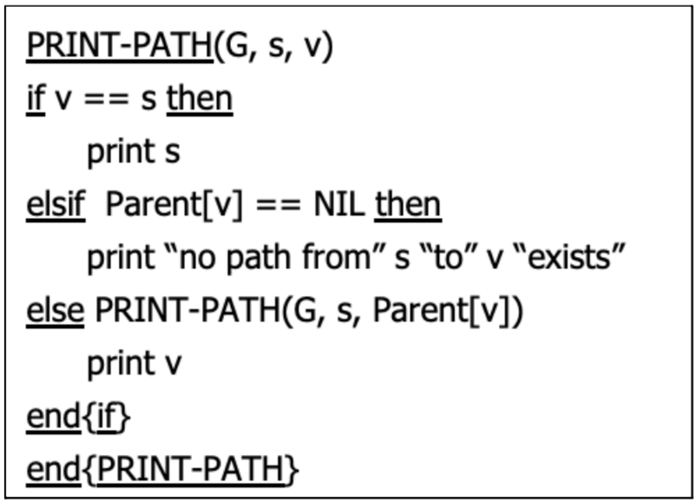


```
while Q != {} loop
    u = DEQUEUE(Q);
    for every v ∈ Adj[u] loop
        if color[v] == white then
            color[v] := gray;
            d[v] = d[u]+1; Parent[v] = u;
            ENQUEUE(Q,v);
        end{if}
    end{loop}
    color[u] = black;
end{loop}
end{BFS}
```

Step 2: Run through the queue

Remember there are two main steps to the algorithm and they are defined in the pseudo-code above:
Data structures: Explain each step of your code using algorithm terminology where possible. This includes the type of data structure you are going to use and how and why you will add and remove elements from the data structure.
Control Structures and Loops. Similarly, explain why you chose a specific type of loop or control statement at a given step.

2.2. Determine the shortest path using results from Breadth-first search.



```
PRINT-PATH(G, s, v)
if v == s then
    print s
elsif  Parent[v] == NIL then
    print "no path from" s "to" v "exists"
else PRINT-PATH(G, s, Parent[v])
    print v
end{if}
end{PRINT-PATH}
```

Once you have the results of your BFS, use it to find the shortest path. See pseudo code above:

2.3. Your S will be AT5G49450 and your V will be AT5G65210