Algorithms and Data Structures for Bioinformatics(BI-GY 7453)

Assignment - 2

Question 1:

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

Part 1: Implement a breadth-first search algorithm using any programming language of your choice.

Remember there are two main steps to the algorithm and they are defined in the pseudo-code below:

Step 1: Initialize the vertices

```
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 2: Run through the queue

```
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 ; Paren[v] = u;
                        ENQUEUE(Q,v);
                end{if}
        end{loop}
        Color[u] = black;
end{loop}
end{BFS}
```

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.

Evaluation Criteria: (10 points)

Code Implementation (5 points):

- Evaluate the correctness of the breadth-first search (BFS) algorithm.
- Check if the algorithm initializes vertices, processes the queue correctly, and updates colors, distances, and parents.

Data Structure Usage (1 point):

- Assess the clarity and appropriateness of the chosen data structure for the queue.

Algorithm Explanation (1 point):

- Evaluate the clarity of the explanation for each step of the BFS algorithm.

Overall Impressions (1 point):

- Provide an overall assessment of the submission, considering both the code implementation and the accompanying explanation.

Remarks:

- Consider providing clear comments within the code to enhance readability.
- Emphasize that the explanation should help someone unfamiliar with the code understand the logic behind each step.
- Encourage the use of descriptive variable names and consistent coding style for better comprehension.

Part 2: Determine the shortest path using results from the Breadth-first search (2 points)

Once you have the results of your BFS, use it to find the shortest path.

See the example below: (You can use this as values for the variables S ,V)

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

Question 2:

In a directed graph with A nodes, given a matrix B of size M x 2 representing M edges such that there is an edge directed from node B[i][0] to node B[i][1], the task is to determine if the graph contains a cycle. Return 1 if a cycle is present, and 0 otherwise. The cycle must include at least two nodes, and there are no self-loops or multiple edges between nodes. The graph may or may not be connected. Nodes are numbered from 1 to A. The solution should handle multiple test cases.

Input Format:

- An integer A representing the number of nodes in the graph.

- A matrix B of size M x 2 representing M edges.

Output Format:

- Return 1 if a cycle is present; otherwise, return 0.

Example Input:

A = 5

B = [[1, 2], [4, 1], [2, 4], [3, 4], [5, 2], [1, 3]]

Example Output:

1

Evaluation Criteria (10 Points):

Code Functionality (4 points):

- Evaluate correctness in identifying cycles in a directed graph.

Comments and Documentation (4 point):

- Emphasize clear and complete comments.
- Assess comments explaining logic, algorithms, and complex code sections.

Efficiency (2 points):

- Evaluate time and space complexity.