



INSTITUTE OF SPACE TECHNOLOGY, ISLAMABAD

BSCS 01 B

**Data Structure and Algorithm Project – Self-Analysis COVID-19 Test Software
(Report)**

**Abeer Ilyas – 200901055
Sukaina Imran – 200901061
Imama Rahmani – 200901007
Muhammad Abdurrehman Ayyub – 200901123**

Batch: BSCS 01

Project Supervisor: Sir Nadeem & Sir Tufail Sajjad

2021 – 22

- **Introduction:**

The project is specifically designed for institutes, in order to keep the COVID-19 result status and make students and teachers aware about their situation. First of all, software asks the user's general data, like name, age, blood group and vaccination information. Afterwards, user is asked 5 questions. As the user answers the questions, a report is formulated using the input data and a record is saved whether the user is COVID-19 positive or COVID-19 negative.

- **Solution Approach:**

Using binary tree data structure to make a structured questionnaire through which program can conclude the result. Using nested if-else command to approach next possible nodes and decision. The program uses the techniques for decision making without involving AI/ML Modules.(Mimicking AI)

- **Project Features:**

Operating System: Windows

IDE(s): VS Code and Visual Studio Community 2019

Data Structure: Binary Tree

Programming Language: Python

Number of code lines: 349

- **Data Structure:**

A Binary Tree starts with a root node. The nodes holds the questions and the edges of the binary tree serve as conditions, leading to the next possible node(question). And the leaf nodes hold the result. Due to nested structure of our project, we have used Binary Tree.

- **Time Complexity:**

- The time complexity of going from one question to another is $O(1)$.
- The time complexity of starting the questionnaire till the result is shown is $O(n)$.

- **Algorithm:**

- Node1 = Are you experiencing COVID-19 symptoms?
- Node2 = Which are you?
- Node3 = Did your result recommend a follow up test?
- Node4 = Is your test positive?
- Node5 = Have you had close contact with someone who's tested positive?
- RN1 = >>> Result: Positive
- RN2 = >>> Result: Negative

If Node1 == Yes:

 If Node2 == Student:

 If Node4 == Yes:

 Then, Print RN1

 Elif Node4 == No:

 If Node5 == Yes:

 Then, Print RN2

 Elif Node5 == No:

 Then, Print RN2

 Elif Node2 == Teacher:

 If Node4 == Yes:

 Then, Print RN1

 Elif Node4 == No:

 If Node5 == Yes:

 Then, Print RN2

 Elif Node5 == No:

 Then, Print RN2

Elif Node1 == No:

 If Node3 == Yes:

 If Node2 == Student:

 If Node4 == Yes:

 Then, Print RN1

 Elif Node4 == No:

 If Node5 == Yes:

 Then, Print RN2

 Elif Node5 == No:

 Then, Print RN2

 Elif Node2 == Teacher:

```

        If Node4 == Yes:

            Then, Print RN1

        Elif Node4 == No:

            If Node5 == Yes:

                Then, Print RN2

            Elif Node5 == No:

                Then, Print RN2

    Elif Node3 == No:

        If Node5== Yes:

            If Node2 == Student:

                Then, Print RN1

            Elif Node2 == Teacher:

                Then, Print RN1

        Elif Node5 == No:

            Then, Print RN2

```

- **Code:**

```

import sys # importing system library
import time # importing time library
import os
os.system('color 0C')

class BinaryTreeNode: # Class for binary tree implementation
    def __init__(self, data): # function for initializing nodes of binary tree
        self.left = None
        self.right = None
        self.data = data

    def Print(self): # function for printing the data in binary tree nodes
        typewriterstyle(self.data)

def typewriterstyle(string): # function for adding type writer printing style with
0.1s delay
    for i in string: # loop for parsing each character of node
        sys.stdout.write(i)
        sys.stdout.flush()
        time.sleep(0.1)

```

```

def typewriterstyle1(string): # function for adding type writer printing style with
0.05s delay
    for i in string:
        sys.stdout.write(i)
        sys.stdout.flush()
        time.sleep(0.05)

if __name__ == '__main__': # Main body
    # storing questions as strings in nodes
    Node1 = BinaryTreeNode('Are you experiencing COVID-19 symptoms?')
    Node2 = BinaryTreeNode('Which are you?')
    Node3 = BinaryTreeNode('Did your result recommend a follow up test?')
    Node4 = BinaryTreeNode('Is your test positive?')
    Node5 = BinaryTreeNode("Have you had close contact with someone who's tested
positive?")
    RN1 = BinaryTreeNode('>>> Result: Positive')
    RN2 = BinaryTreeNode('>>> Result: Negative')
    start = time.time() # starting timer

    Opening = "Self-Analysis COVID-19 Test Software Report"

    typewriterstyle('Enter your name: ')
    Name = input()
    typewriterstyle('Enter your age: ')
    Age = input()
    typewriterstyle('Enter your blood group: ')
    BG = input()
    typewriterstyle1('Are you vaccinated or not?\nType "Yes" or "No": ')
    Vaccination = input()
    typewriterstyle('\n')
    Node1.Print()
    time_complexity1_start = time.time() # starting timer time to calculate one node
traversing of the tree
    typewriterstyle1('\nType "Yes" for Yes.\nType "No" for No.\nEnter: ')
    opt = input()
    # Main branching
    if opt == 'Yes' or opt == 'YES' or opt == 'yes' or opt == 'y' or opt == 'Y':
        typewriterstyle('>>> GET TESTED!(Diagnostic)\n')
        Node2.Print()
        time_complexity1_end = time.time() # ending timer of node traversal time
checking timer
        typewriterstyle1('\nType "Student" or "Teacher": ')
        opt1 = input()
        # Secondary branching
        if opt1 == 'Student':

```

```

        typewriterstyle(">>> Go to university's diagnostic center.\n")
Node4.Print()
typewriterstyle1('\nType "Yes" for Yes.\nType "No" for No.\nEnter: ')
opt = input()
# Tertiary branching
if 'Yes' == opt or opt == 'YES' or opt == 'yes' or opt == 'y' or opt ==
'Y':
        typewriterstyle('>>> Isolate yourself from others and consider the
best treatment.\n')

        # Printing data for reporting
        print('\n\n')
        print(Opening.center(100, "*"))
        print('Name: ', Name)
        print('Age: ', Age)
        print('Blood Group: ', BG)
        print('Vaccination: ', Vaccination)
        RN1.Print()

elif opt == 'No' or opt == 'NO' or opt == 'no' or opt == 'n' or opt ==
'N':
        Node5.Print()
        typewriterstyle1('\nType "Yes" for Yes.\nType "No" for No.\nEnter:
')

        opt = input()
        # Quaternary branching
        if 'Yes' == opt or opt == 'YES' or opt == 'yes' or opt == 'y' or opt
== 'Y':
                typewriterstyle('>>> Still be safe and ensure social
distancing.\n')

                print('\n\n')
                print(Opening.center(100, "*"))
                print('Name: ', Name)
                print('Age: ', Age)
                print('Blood Group: ', BG)
                print('Vaccination: ', Vaccination)
                RN2.Print()

elif opt == 'No' or opt == 'NO' or opt == 'no' or opt == 'n' or opt
== 'N':
        typewriterstyle('>>> Still be safe.\n')

        print('\n\n')
        print(Opening.center(100, "*"))
        print('Name: ', Name)
        print('Age: ', Age)
        print('Blood Group: ', BG)

```

```

        print('Vaccination: ', Vaccination)
        RN2.Print()

    else:
        typewriterstyle('No such option exists.\n')

else:
    typewriterstyle('No such option exists.\n')

elif opt1 == 'Teacher':
    typewriterstyle(">>> Visit the nearest health center or hospital as soon
as possible.\n")
    Node4.Print()
    typewriterstyle1('\nType "Yes" for Yes.\nType "No" for No.\nEnter: ')
    opt = input()
    # Secondary branching
    if opt == 'Yes' or opt == 'YES' or opt == 'yes' or opt == 'y' or opt ==
'Y':
        typewriterstyle('>>> Take a leave and follow medication.\n')

        print('\n\n')
        print(Opening.center(100, "*"))
        print('Name: ', Name)
        print('Age: ', Age)
        print('Blood Group: ', BG)
        print('Vaccination: ', Vaccination)
        RN1.Print()

    elif opt == 'No' or opt == 'NO' or opt == 'no' or opt == 'n' or opt ==
'N':
        Node5.Print()
        typewriterstyle1('\nType "Yes" for Yes.\nType "No" for No.\nEnter:
')

        opt = input()
        # Tertiary branching
        if 'Yes' == opt or opt == 'YES' or opt == 'yes' or opt == 'y' or opt
== 'Y':
            typewriterstyle1('>>> Still be safe and ensure social
distancing.\n')

            print('\n\n')
            print(Opening.center(100, "*"))
            print('Name: ', Name)
            print('Age: ', Age)
            print('Blood Group: ', BG)
            print('Vaccination: ', Vaccination)
            RN2.Print()

```

```

elif opt == 'No' or opt == 'NO' or opt == 'no' or opt == 'n' or opt
== 'N':

    typewriterstyle('>>> Still be safe.\n')

    print('\n\n')
    print(Opening.center(100, "*"))
    print('Name: ', Name)
    print('Age: ', Age)
    print('Blood Group: ', BG)
    print('Vaccination: ', Vaccination)
    RN2.Print()

    else:
        typewriterstyle('No such option exists.\n')

    else:
        typewriterstyle('No such option exists.\n')

    else:
        typewriterstyle('No such option exists.\n')

elif opt == 'No' or opt == 'NO' or opt == 'no' or opt == 'n' or opt == 'N':
    typewriterstyle('>>> GET TESTED!(Asymptomatic)\n')
    Node3.Print()
    time_complexity1_end = time.time() # ending timer of node traversal time
checking timer
    typewriterstyle1('\nType "Yes" for Yes.\nType "No" for No.\nEnter: ')
    opt = input()
    # Secondary branching
    if opt == 'Yes' or opt == 'YES' or opt == 'yes' or opt == 'y' or opt == 'Y':
        Node2.Print()
        typewriterstyle1('\nType "Student" or "Teacher": ')
        opt1 = input()
        # Tertiary branching
        if opt1 == 'Student':
            Node4.Print()
            typewriterstyle1('\nType "Yes" for Yes.\nType "No" for No.\nEnter:
')
            opt = input()
            # Quaternary branching
            if opt == 'Yes' or opt == 'YES' or opt == 'yes' or opt == 'y' or opt
== 'Y':

                typewriterstyle('>>> Isolate yourself from others and consider
the best treatment.\n')

                print('\n\n')
                print(Opening.center(100, "*"))

```



```

        print('Name: ', Name)
        print('Age: ', Age)
        print('Blood Group: ', BG)
        print('Vaccination: ', Vaccination)
        RN1.Print()

    elif opt == opt == 'No' or opt == 'NO' or opt == 'no' or opt == 'n'
or opt == 'N':
        Node5.Print()
        typewriterstyle1('\nType  "Yes"  for  Yes.\nType  "No"  for
No.\nEnter: ')
        opt = input()
        # Quaternary branching
        if 'Yes' == opt or opt == 'YES' or opt == 'yes' or opt == 'y' or
opt == 'Y':
            typewriterstyle('>>> Still be safe and ensure social
distancing.\n')

            print('\n\n')
            print(Opening.center(100, "*"))
            print('Name: ', Name)
            print('Age: ', Age)
            print('Blood Group: ', BG)
            print('Vaccination: ', Vaccination)
            RN2.Print()

        elif opt == 'No' or opt == 'NO' or opt == 'no' or opt == 'n' or
opt == 'N':
            typewriterstyle('>>> Still be safe.\n')

            print('\n\n')
            print(Opening.center(100, "*"))
            print('Name: ', Name)
            print('Age: ', Age)
            print('Blood Group: ', BG)
            print('Vaccination: ', Vaccination)
            RN2.Print()

        else:
            typewriterstyle('No such option exists.\n')

    else:
        typewriterstyle('No such option exists.\n')

elif opt1 == 'Teacher':
    Node4.Print()
    typewriterstyle1('\nType  "Yes"  for  Yes.\nType  "No"  for  No.\nEnter:
')

```

```

        opt = input()
        # Tertiary branching
        if opt == 'Yes' or opt == 'YES' or opt == 'yes' or opt == 'y'
or opt == 'Y':
            typewriterstyle('>>> Take a leave and follow medication.\n')

            print('\n\n')
            print(Opening.center(100, "*"))
            print('Name: ', Name)
            print('Age: ', Age)
            print('Blood Group: ', BG)
            print('Vaccination: ', Vaccination)
            RN1.Print()

        elif opt == 'No' or opt == 'NO' or opt == 'no' or opt == 'n' or opt
== 'N':
            Node5.Print()
            typewriterstyle1('\nType  "Yes"  for  Yes.\nType  "No"  for
No.\nEnter: ')
            opt = input()
            # Quaternary branching
            if 'Yes' == opt or opt == 'YES' or opt == 'yes' or opt == 'y' or
opt == 'Y':
                typewriterstyle('>>> Still be safe and ensure social
distancing.\n')

                print('\n\n')
                print(Opening.center(100, "*"))
                print('Name: ', Name)
                print('Age: ', Age)
                print('Blood Group: ', BG)
                print('Vaccination: ', Vaccination)
                RN2.Print()

            elif opt == 'No' or opt == 'NO' or opt == 'no' or opt == 'n' or
opt == 'N':
                typewriterstyle('>>> Still be safe.\n')

                print('\n\n')
                print(Opening.center(100, "*"))
                print('Name: ', Name)
                print('Age: ', Age)
                print('Blood Group: ', BG)
                print('Vaccination: ', Vaccination)
                RN2.Print()

            else:
                typewriterstyle('No such option exists.\n')

```

```

        else:
            typewriterstyle('No such option exists.\n')

    else:
        typewriterstyle('No such option exists.\n')

elif opt == 'No' or opt == 'NO' or opt == 'no' or opt == 'n' or opt == 'N':
    Node5.Print()
    typewriterstyle1('\nType "Yes" for Yes.\nType "No" for No.\nEnter: ')
    opt = input()
    # Tertiary branching
    if opt == 'Yes' or opt == 'YES' or opt == 'yes' or opt == 'y' or opt ==
'Y':
        Node2.Print()
        typewriterstyle1('\nType "Student" or "Teacher": ')
        opt1 = input()
        # Quaternary branching
        if opt1 == 'Student':
            typewriterstyle('>>> Isolate yourself from others and consider
the best treatment.\n')

            print('\n\n')
            print(Opening.center(100, "*"))
            print('Name: ', Name)
            print('Age: ', Age)
            print('Blood Group: ', BG)
            print('Vaccination: ', Vaccination)
            RN1.Print()

        elif opt1 == 'Teacher':
            typewriterstyle('>>> Take a leave and follow medication.\n')

            print('\n\n')
            print(Opening.center(100, "*"))
            print('Name: ', Name)
            print('Age: ', Age)
            print('Blood Group: ', BG)
            print('Vaccination: ', Vaccination)
            RN1.Print()

        else:
            typewriterstyle('No such option exists.\n')

elif opt == 'No' or opt == 'NO' or opt == 'no' or opt == 'n' or opt ==
'N':
    typewriterstyle('>>> Still be safe.\n')

```

```

        print('\n\n')
        print(Opening.center(100, "*"))
        print('Name: ', Name)
        print('Age: ', Age)
        print('Blood Group: ', BG)
        print('Vaccination: ', Vaccination)
        RN2.Print()

    else:
        typewriterstyle('No such option exists.\n')

else:
    typewriterstyle('No such option exists.\n')

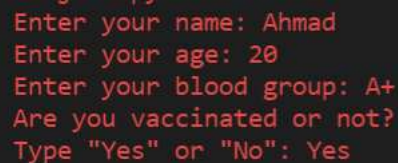
else:
    typewriterstyle('No such option exists.\n')

end = time.time()
# Printing time taken for single node traversal
typewriterstyle('\n\n\n>>> Node Traversal time: ')
print(time_complexity1_end - time_complexity1_start)
# Printing time taken for whole traversing traversal
typewriterstyle('\n\n\n>>> Tree Traversal time: ')
print(end - time_complexity1_start)
# Printing time taken for program runtime
typewriterstyle('\n\n\n>>> Program runtime: ')
print(end - start)

os.system("PAUSE")

```

- **Screenshots:**

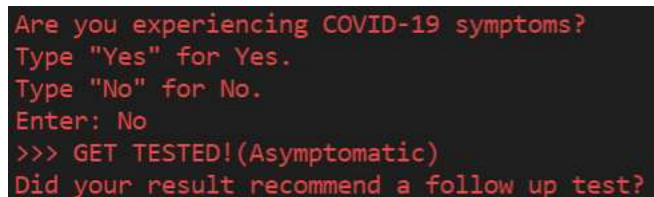


```

Enter your name: Ahmad
Enter your age: 20
Enter your blood group: A+
Are you vaccinated or not?
Type "Yes" or "No": Yes

```

General Data Input



```

Are you experiencing COVID-19 symptoms?
Type "Yes" for Yes.
Type "No" for No.
Enter: No
>>> GET TESTED!(Asymptomatic)
Did your result recommend a follow up test?

```

Questionnaire

```
*****Self-Analysis COVID-19 Test Software Report*****  
Name: Abeer Ilyas  
Age: 20  
Blood Group: A+  
Vaccination: Yes  
>>> Result: Negative
```

Report

```
>>> Node Traversal time: 9.449419736862183  
  
>>> Tree Traversal time: 45.04087972640991  
  
>>> Program runtime: 68.85561394691467
```

Time calculation for analyzing time complexity

- **Applications:**

- This software can be used to analyze the COVID-19 status of each person in any sector such as:
 - Universities
 - Offices
 - Public Places
 - Home
- We can alternate the program in such a way that we can create software for decision making and questionnaire.

- **Future upgradations:**

File handling can be implemented to save record permanently. Secondly, AI Decision Tree can be implemented in future upgradation for analyzing the data of mass number of users. This can be used to analyze the COVID-19 positivity ratio in a specific region and makes decision making easier for administration.