VIT®

Vellore Institute of Technology
(Deemed to be University under section 3 of UGC Act, 1956)

BUILDING INTERACTIVE TOOLS TO TEACH BIOINSPIRED ALGORITHMS

PROJECT FOR:

BIT1028 – Bio -Inspired design

Under the guidance of

Dr. DEBASISH MISHRA

ANIKET KUMAR (18BCE0101)

ANURAG KASHYAP (18BCE0776)

SAPAN SHRIVASTAVA (18BCE0779)

DECLARATION

WE HEREBY DECLARE THAT THE THESIS ENTITTLED "BUILDING INTERACTIVE TOOLS FOR TEACHING BIOINSPIRED ALGORITHMS " SUBMITTED BY US, FOR THE AWARD OF THE DEGREE OF *BACHELOR OF TECHNOLOGY IN COMPUTER SCIENCE AND ENGINEERING* TO VIT IS A RECORD OF BONAFIDE WORK CARRIED OUT BY ME UNDER THE SUPERVISION OF DR.DEBASISH MISHRA .

.

Place: Vellore

Date:

05-06-2020

ANIKET KUMAR
ANURAG KASHYAP
SAPAN SHRIVASTAVA
Signature of the Candidate

ACKNOWLEDGMENT

We take this opportunity to express our heartfelt gratitude to our guide Dr. Debasish Mishra whose able guidance helped me in the development of proposed system for my problem statement. We shall ever remain indebted for his meticulous guidance, constructive criticism, clear thinking, keen interest, constant encouragement and forbearing right from the beginning of this research to it’s completion. Finally, we would like to thank VIT University for incorporating this project as a part of our curriculum and encouraging our creativity

SUMMARY OF REPORT

The aim of this project is to create easy, accessible and affective tools for teaching basic biological algorithms. Biological algorithms have been around for a very long time and have provided solutions to a lot of phenomena. However, their contribution is little known to those not involved in the biological field. This could be attributed to the complexity of the algorithms or the lack of understanding the basic biological concepts. Therefor this project aims on creating a tools for such people to understand biological algorithms and their functioning in a very simple way without having to be aware of the concepts. These tools are coded in a way to deconstruct and show how the basic working of different algorithms. Most existing teaching tools are primitive and expensive and require a basic knowledge of the fundamentals. So, this tool fixes all these issues by making learning accessible, cheap and easily understandable. These application will be developed using java and python. This application is the first of its kind because of which the approach being followed would be very preliminary and simple. However, it has a lot of scope and be developed extensively to make it more robust and effective. This application aims on explaining both the physical and theoretical aspects of the biological algorithms. As of know the algorithms have been categorized into categories based on their functionality including genetic and optimization algorithms .

TABLE OF CONTENTS

INTRODUCTION

Online applications to study subjects are becoming more and more common these days. However, applications specific a certain niche area have not really been worked on or thought into so far. Especially there is no existing application to study the working of algorithms let alone biological algorithms.

Biological algorithms were developed to gain an understanding the functioning of different biological phenomena. They are very specific and almost entirely nature based. They at times could be very complex as they deal with the intricacies of how our biosphere works. Unlike regular algorithms in other areas they may sometimes not even have finite steps. They are of various different types like optimization based algorithms, genetic algorithms etc.

Recently more and more people have been showing an interest in learning these biological algorithms. This rising interest could be attributed to the app developers, scientists and technology experts finding applications of them in developing various new advanced technologies. Experts from the area of mechanics and constructions have also been trying to understand the biological algorithms to make machines and structures which are much more sturdy and resilient.

Therefore if an application was to be developed specifically to teach and deconstruct these algorithms it might prove to be very helpful to everyone interested in understanding the biological algorithms. This application aims on developing many minor applications specific to each algorithm launched through a website and maybe a phone application in future . These minor applications mainly aim to simulate the working different algorithms. This project main aims on creating a basic prototype of the application as the actual application might need a lot more man power and technological expertise to be developed o a full extent .

This application mainly aims on making study of algorithms easy and accessible to everyone .

1.1 Objective

This project will take basic biological algorithms and convert them into interactive web applications which can be used by anyone. It will make the users experience very productive and effective using a game like method of teaching. The aim of this project is to create easy , accessible and affective tools for teaching basic biological algorithms .These teaching tools are to be developed using computer code (mainly python, JavaScript).

The major objective here is to make a very easy accessible way to teach algorithms to students/tutors/web developers etc. who might not have little to no idea of biology any related concepts. There are different categories of algorithms making a process of understanding these algorithms which makes it much complex to understand so at its full capacity this project aims on segregating as per their function and purpose.

This project is just a prototype to further work on and make it functional maybe used form teaching other algorithms of various other categories and disciplines. Another objective of this project is to make this very user friendly and free of cost for everyone so that its accessible to everyone.

So far teaching applications have not really been able to use concepts of deconstructive teaching techniques employing latest technologies like machine learning and A.I. .Also this application aims on using a hands on approach to understand the projects instead of mugging up concepts.

1.2 Motivation

The motivation mainly came from the lack of area specific teaching tools especially the lack of good teaching tools for biological algorithms or algorithms in general. This application aims on teaching algorithms in a comprehensive step by step manor. The learners can get a hands on

practical experience which will not only improve the chances of them understanding the but also makes the process much more interesting and fun overall .

The idea primarily originated form a conversation with my project guide . The fact that there is no tool for specifically teaching algorithms out there and specifically there is no tool to help you learn biological algorithms in specific .

Another major motivating factor is to provide a very pocket friendly source for students especially to learn these algorithms . Apart from this they would also get to learn it hands on and understand the intricacies of each of the algorithm while going through the coded procedure

1.3 Background

An algorithm is a process or set of rules to be followed in calculation or other problem solving operations. They are used as blue prints to design applications and to go about designing projects systematically. They play a major role in every field to build systematic. Biological algorithms were created to deal with the problems which came up while understanding different areas of biology. The projects which specifically employ these algorithms are those involving bio inspired design and genetics. However recently it has been noted that these algorithms have more implementations outside the field of biology.

So more and more software / tech designers and students, are showing inclination towards studying and understanding these algorithms. however, it is harder for them to gauge these concepts due to their lack of basic knowledge necessary. That is where this application comes into play. This application simplifies the process of learning for the interested individuals.

According to most surveys these days it has been suggested that online learning using visual aids has proven to be very effective. This application aims on understanding algorithms using the concept of simulation to understand biological algorithms.

The main idea here is to make something open source and accessible to everyone to study and learn the working of biological algorithms.

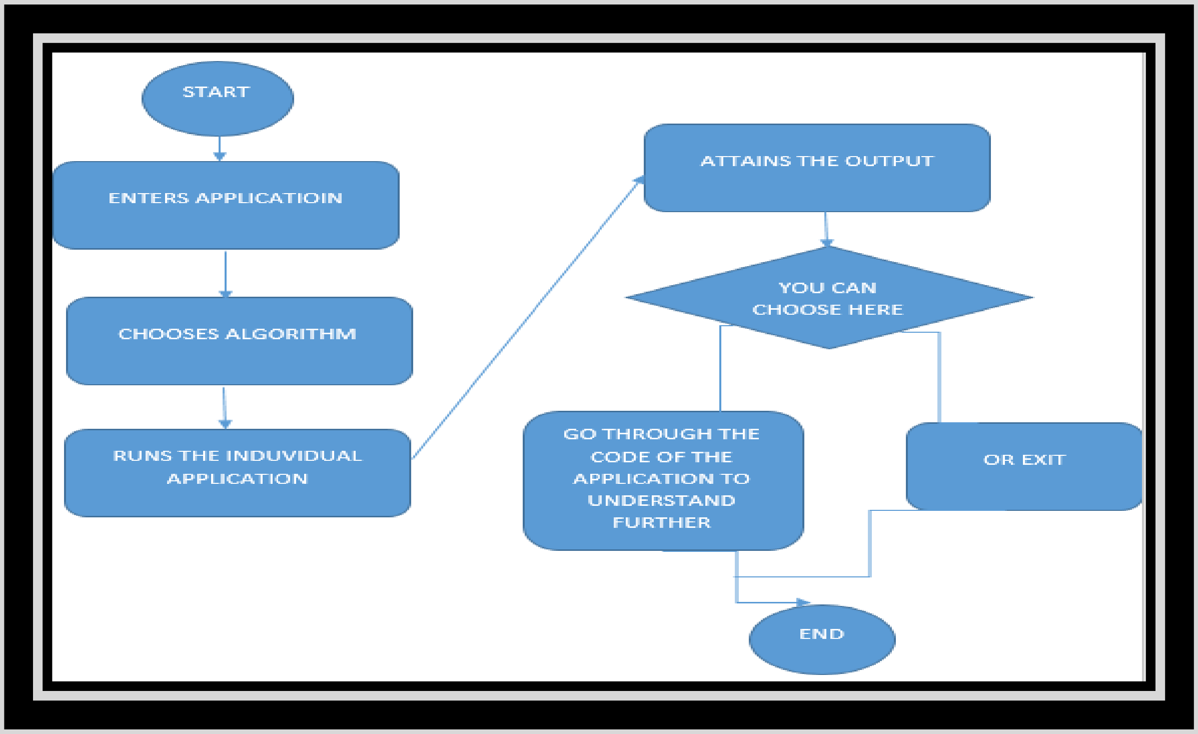
1) PROJECT DESCRIPTION AND GOALS

2.1 Project Description

The project is a web application to teach Biological/Bioinspired algorithms using interactive web applications. These applications would be developed using python/JavaScript. It would basically consist of a number of sub applications which help the user to understand the algorithms hands on , step by step . They can also go through the code of the application to understand the overall working.

The application is categories the algorithms based on their functionally and utility presently eventually as more algorithms are included and the complexity increases the atomicity requiring more categorization.

Presently the aim of the project is to develop the prototype of the web application as it is extremely code intensive. Eventually the application would be made accessible to everyone they would even have a chance to not only learn but also add on or edit the applications code to make the simulations more effective, better and easily understandable. This application also aims on being free and not maintenance intensive. It aims on putting a spin on a regular teaching application which are not as interactive and also might be expensive.


START
ENTERS APPLICATIOIN
CHOOSES ALGORITHM
RUNS THE INDUVIDUAL APPLICATION
ATTAINS THE OUTPUT
YOU CAN CHOOSE HERE
GO THROUGH THE CODE OF THE APPLICATION TO UNDERSTAND FURTHER
OR EXIT
END

<u>LITERATURE SURVEY</u>

3.1. Survey of the Existing Models/Work

Over performing an extensive literature survey it has been noted that there is little to no material available on this approach. It is very evident that this is a rather new approach. No teaching applications so far specialize in teaching algorithms. Literature pertaining to this topic is fairly limited posing a much bigger challenge. The only literature I could find was based on the development of interactive web applications

3.2. Summary/Gaps identified in the Survey

The gaps that were observed mainly here are the lack of exploration into the importance of teaching and learning algorithms. It was found that as of now there is no available teaching tools that has the capabilities of this application. Therefor there is no existing model to work off of .In conclusion a model has to be developed form scratch in order to create this tool .

3) TECHNICAL SPECIFICATION

4.1 describing of different elements of the project

Algorithm: -An algorithm is a set of steps followed to solve a problem. Algorithms are generally primarily in the field of computer science but, of more recently this approach is being adopted by other fields like biology.

Bioinspired algorithm: -They are algorithms that are derived from the working of various natural phenomena which can then be applied elsewhere. They are a component of Bioinspired computing which is a combinational study of social behaviors, connectionism and emergence. These algorithms heavily rely on concepts of Mathematics, Biology and Computer Science.

Computer Application: - It is an application built using different programing languages to deliver a certain type of service. Services could range from health care to Retail to Teaching. In this particular case our application is a teaching based .

Interactive Teaching Tools :-These are computer applications which provide the service of online based learning .The difference being that the students can be an active part of the learning experience instead of being just being a distant spectator
.

Genetic Algorithms :-A genetic algorithm is a method of analysis that is inspired by Charles Darwin's theory of natural evolution. This algorithm analyze a given population to determine its fitness as a result of parameters like natural selection, mutation and crossover. It is a biological algorithms

Optimization Algorithms :-These algorithms minimize / remove redundancy inn a given data set
. They help understand the structural and functional aspects of the data given
in this case biological (nucleotide bases , amino acids).

SCHEDULE TASKS AND ALGORITHMS

The aim of this project is to initially to generate individual applications and then if time permits to create a comprehensive website to host these applications. Since this is a very extensive project only a basic prototype can be generated given the time span given and the resources available. In order to generate this prototype four different types of algorithms were choose primarily they include :-

They are :-

SmithWaterman Algorithm

The Smith–Waterman algorithm performs local sequence alignment, that is, for determining similar regions between two strings of nucleic acid sequences or protein sequences. Instead of looking at the entire sequence, the Smith–Waterman algorithm compares segments of all possible lengths and optimizes the similarity measure.

APPLICATIONS

FPGA

CRAY demonstrated acceleration of the Smith–Waterman algorithm using a reconfigurable computing platform based on FPGA chips, with results showing up to 28x speed-up over standard microprocessorbased solutions. Another FPGA-based version of the Smith–Waterman algorithm shows FPGA (Virtex-4) speedups up to 100x over a 2.2 GHz Opteron processor. The Time Logic De Cypher and Code Quest systems also accelerate Smith–Waterman and Frame search using PCIe FPGA cards.

SIMD

In 2000, a fast implementation of the Smith–Waterman algorithm using the SIMD technology available in Intel Pentium MMX processors and similar technology was described in a publication by Rognes and Seeberg. In contrast to the Wozniak (1997) approach, the new implementation was based on vectors parallel with the query sequence, not diagonal vectors.

Ant Swarm Algorithm

In computer science and operations research, the ant colony optimization algorithm (ACO) is a probabilistic technique for solving computational problems which can be reduced to finding good paths through graphs. Artificial Ants stand for multi-agent methods inspired by the behaviour of real ants. As an example, Ant colony optimization, is a class of optimization algorithms modelled on the actions of an ant colony. Artificial 'ants’ locate optimal solutions by moving through a parameter space representing all possible solutions. Real ants lay down pheromones directing each other to resources while exploring their environment. The simulated 'ants' similarly record their positions and the quality of their solutions, so that in later simulation iterations more ants locate better solutions

APPLICATIONS

1) The first ACO algorithm was called the ant system and it was aimed to solve the travelling salesman problem, in which the goal is to find the shortest round-trip to link a series of cities. The general algorithm is relatively simple and based on a set of ants, each making one of the possible round-trips along the cities. At each stage, the ant chooses to move from one city to another according to some rules:
a) It must visit each city exactly once
b) A distant city has less chance of being chosen
c) The more intense the pheromone trail laid out on an edge between two cities, the greater the probability that that edge will be chosen.
d) Having completed its journey, the ant deposits more pheromones on all edges it traversed, if the journey is short
e) After each iteration, trails of pheromones evaporate.

2) Another application is of Vehicle Routing System. Many problems are solved using this technique like:
• Capacitated vehicle routing problem.
• Split delivery vehicle routing problem.
• Vehicle routing problem with pick-up and delivery problem.
• Multi-depot vehicle routing problem.
• Period vehicle routing problem.

6.1) TASKS: -

So, the tasks that have to be taken care are:-

- to research about the chosen algorithms
- Break them on the basis of their principle
- Then convert the algorithms into a coded form using primarily python and Java if required
- While building these algorithms we need to take care of the fact that the algorithms need to be sell sufficient
- Once the design phase is done the testing phase begins.

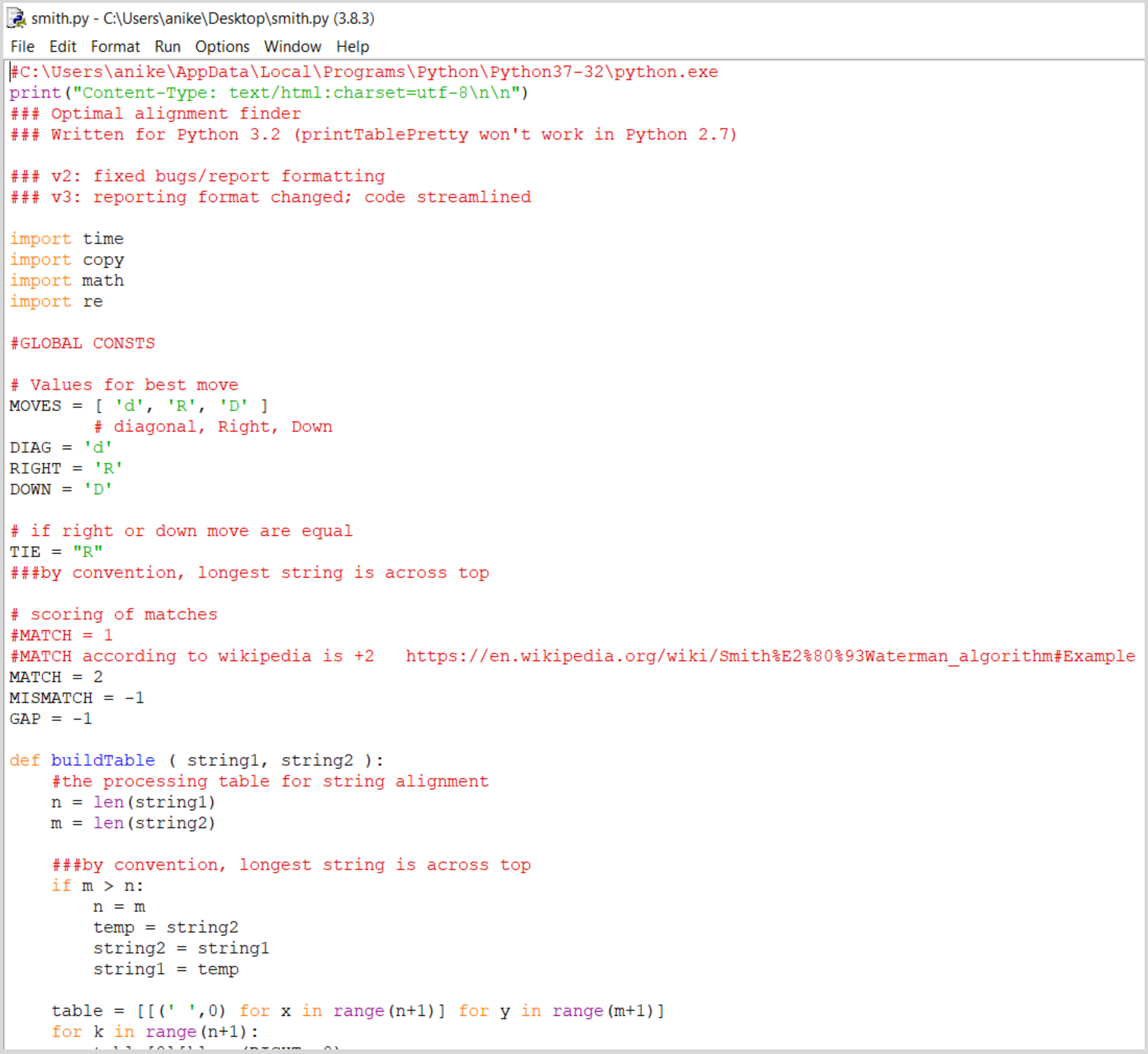
- While in the testing phase the codes need to be put through immense screening to remove not just the errors but also the excess unwanted loops which reduce the efficiency of the algorithms.
- Once the applications have been developed we can start to work on the website part of project.
- The website could mainly be developed using java and python for the backend and Html CSS and basic java script for the front end . We could also use pre made online templates to develop the prototype websites.
- However the primary more important task is to work on the applications and then go to the website if time permits

2) PROJECT DEMONSTRATION

The project primarily employed python and java to realize most of the project .

Most of the code was python based and was coded on and compiled using IDLE . IDLE hence will also be used for demonstrating the project

- The Layout and the code looks somewhat like this. Below is the layout of the Ant colony Code.


smith.py - C:\Users\anike\Desktop\smith.py (3.8.3)
File Edit Format Run Options Window Help

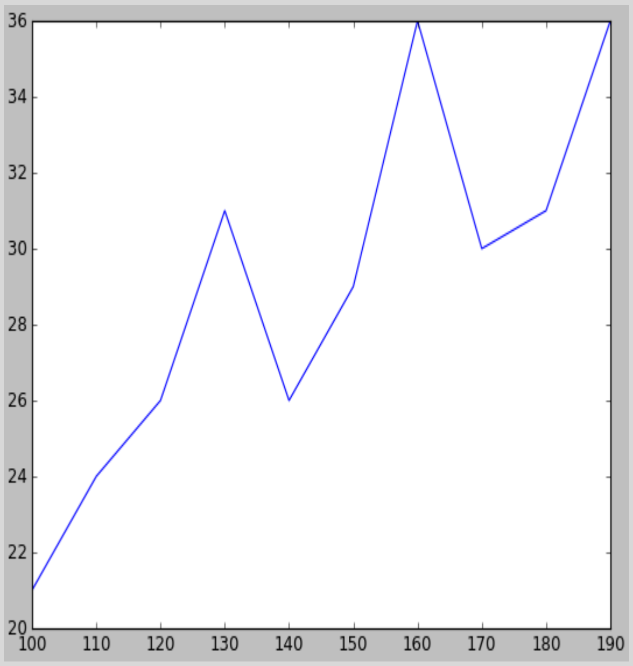
```
#C:\Users\anike\AppData\Local\Programs\Python\Python37-32\python.exe
print("Content-Type: text/html:charset=utf-8\n\n")
### Optimal alignment finder
### Written for Python 3.2 (printTablePretty won't work in Python 2.7)

### v2: fixed bugs/report formatting
### v3: reporting format changed; code streamlined

import time
import copy
import math
import re

#GLOBAL CONSTS

# Values for best move
MOVES = [ 'd', 'R', 'D' ]
        # diagonal, Right, Down
DIAG = 'd'
RIGHT = 'R'
DOWN = 'D'

# if right or down move are equal
TIE = "R"
###by convention, longest string is across top

# scoring of matches
#MATCH = 1
#MATCH according to wikipedia is +2   https://en.wikipedia.org/wiki/Smith%E2%80%93Waterman_algorithm#Example
MATCH = 2
MISMATCH = -1
GAP = -1

def buildTable ( string1, string2 ):
    #the processing table for string alignment
    n = len(string1)
    m = len(string2)

    ###by convention, longest string is across top
    if m > n:
        n = m
        temp = string2
        string2 = string1
        string1 = temp

    table = [[(' ',0) for x in range(n+1)] for y in range(m+1)]
    for k in range(n+1):
```

You can notice that numpy and matplotlib have been used to generate the deesierd outfit and this is how the result of the compilation looks

A screenshot of Co- ordinates generated through nine experiments

Fig. 3

The Plot developed tracing the path of the ants based on the co-ordinates generated

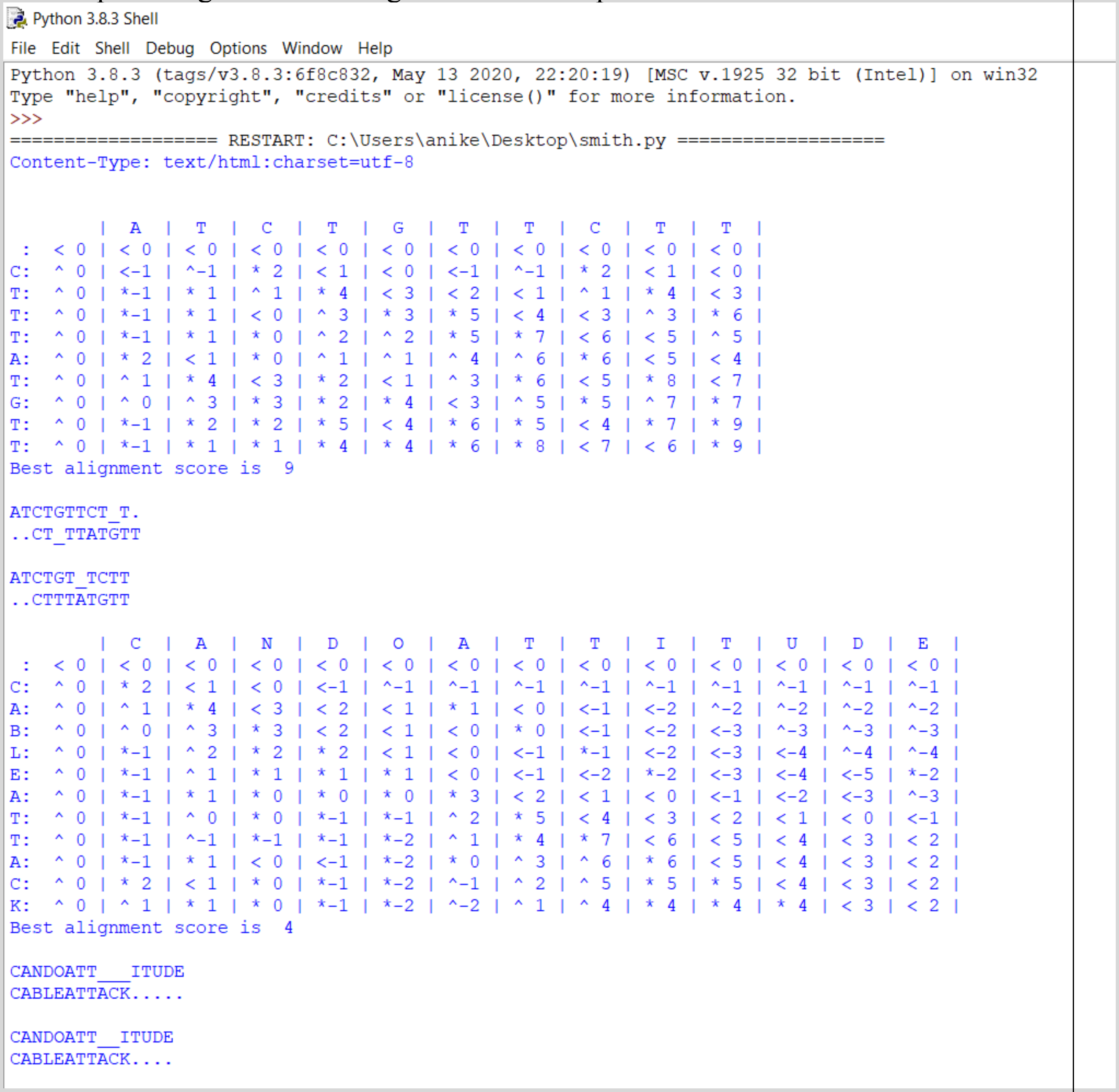


The rest of the python demonstrations follow a similar approach and have been mentioned in result with the algorithms and theoretical background of each of them .

There was only one java based code and the initial plan was to code using NetBeans . However due to some technical difficulties the code had to be written in NotePad and compiled using an online compiler .

The demonstration is as

The output was generated through an online compiler :-



```
Python 3.8.3 (tags/v3.8.3:6f8c832, May 13 2020, 22:20:19) [MSC v.1925 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
================== RESTART: C:\Users\anike\Desktop\smith.py ===================
Content-Type: text/html:charset=utf-8


        |  A  |  T  |  C  |  T  |  G  |  T  |  T  |  C  |  T  |  T  |
 :  < 0 | < 0 | < 0 | < 0 | < 0 | < 0 | < 0 | < 0 | < 0 | < 0 | < 0 |
C:  ^ 0 | <-1 | ^-1 | * 2 | < 1 | < 0 | <-1 | ^-1 | * 2 | < 1 | < 0 |
T:  ^ 0 | *-1 | * 1 | ^ 1 | * 4 | < 3 | < 2 | < 1 | ^ 1 | * 4 | < 3 |
T:  ^ 0 | *-1 | * 1 | < 0 | ^ 3 | * 3 | * 5 | < 4 | < 3 | ^ 3 | * 6 |
T:  ^ 0 | *-1 | * 1 | * 0 | ^ 2 | ^ 2 | * 5 | * 7 | < 6 | < 5 | ^ 5 |
A:  ^ 0 | * 2 | < 1 | * 0 | ^ 1 | ^ 1 | ^ 4 | ^ 6 | * 6 | < 5 | < 4 |
T:  ^ 0 | ^ 1 | * 4 | < 3 | * 2 | < 1 | ^ 3 | * 6 | < 5 | * 8 | < 7 |
G:  ^ 0 | ^ 0 | ^ 3 | * 3 | * 2 | * 4 | < 3 | ^ 5 | * 5 | ^ 7 | * 7 |
T:  ^ 0 | *-1 | * 2 | * 2 | * 5 | < 4 | * 6 | * 5 | < 4 | * 7 | * 9 |
T:  ^ 0 | *-1 | * 1 | * 1 | * 4 | * 4 | * 6 | * 8 | < 7 | < 6 | * 9 |
Best alignment score is  9

ATCTGTTCT_T.
..CT_TTATGTT

ATCTGT_TCTT
..CTTTATGTT

        |  C  |  A  |  N  |  D  |  O  |  A  |  T  |  T  |  I  |  T  |  U  |  D  |  E  |
 :  < 0 | < 0 | < 0 | < 0 | < 0 | < 0 | < 0 | < 0 | < 0 | < 0 | < 0 | < 0 | < 0 | < 0 |
C:  ^ 0 | * 2 | < 1 | < 0 | <-1 | ^-1 | ^-1 | ^-1 | ^-1 | ^-1 | ^-1 | ^-1 | ^-1 | ^-1 |
A:  ^ 0 | ^ 1 | * 4 | < 3 | < 2 | < 1 | * 1 | < 0 | <-1 | <-2 | ^-2 | ^-2 | ^-2 | ^-2 |
B:  ^ 0 | ^ 0 | ^ 3 | * 3 | < 2 | < 1 | < 0 | * 0 | <-1 | <-2 | <-3 | ^-3 | ^-3 | ^-3 |
L:  ^ 0 | *-1 | ^ 2 | * 2 | * 2 | < 1 | < 0 | <-1 | *-1 | <-2 | <-3 | <-4 | ^-4 | ^-4 |
E:  ^ 0 | *-1 | ^ 1 | * 1 | * 1 | * 1 | < 0 | <-1 | <-2 | *-2 | <-3 | <-4 | <-5 | *-2 |
A:  ^ 0 | *-1 | * 1 | * 0 | * 0 | * 0 | * 3 | < 2 | < 1 | < 0 | <-1 | <-2 | <-3 | ^-3 |
T:  ^ 0 | *-1 | ^ 0 | * 0 | *-1 | *-1 | ^ 2 | * 5 | < 4 | < 3 | < 2 | < 1 | < 0 | <-1 |
T:  ^ 0 | *-1 | ^-1 | *-1 | *-1 | *-2 | ^ 1 | * 4 | * 7 | < 6 | < 5 | < 4 | < 3 | < 2 |
A:  ^ 0 | *-1 | * 1 | < 0 | <-1 | *-2 | * 0 | ^ 3 | ^ 6 | * 6 | < 5 | < 4 | < 3 | < 2 |
C:  ^ 0 | * 2 | < 1 | * 0 | *-1 | *-2 | ^-1 | ^ 2 | ^ 5 | * 5 | * 5 | < 4 | < 3 | < 2 |
K:  ^ 0 | ^ 1 | * 1 | * 0 | *-1 | *-2 | ^-2 | ^ 1 | ^ 4 | * 4 | * 4 | * 4 | < 3 | < 2 |
Best alignment score is  4

CANDOATT___ITUDE
CABLEATTACK.....

CANDOATT__ITUDE
CABLEATTACK....
```

RESULTS AND DISCUSSIONS:

Here are the results and discussions of each of the tools individually ACO (Ant Colony Optimization)

Assumptions/Parameters

- Assumption The initial pheromone level of each edge is assumed to be 1, to avoid the problem not selecting a path due to zero pheromone.
- Also the ants follow a strategy to choose a path randomly with a chance of 0.1.
- Since all ants start randomly from a postition at any vertex, the best path does not account for the same cyclic paths that another ant starting from another vertex might have taken.

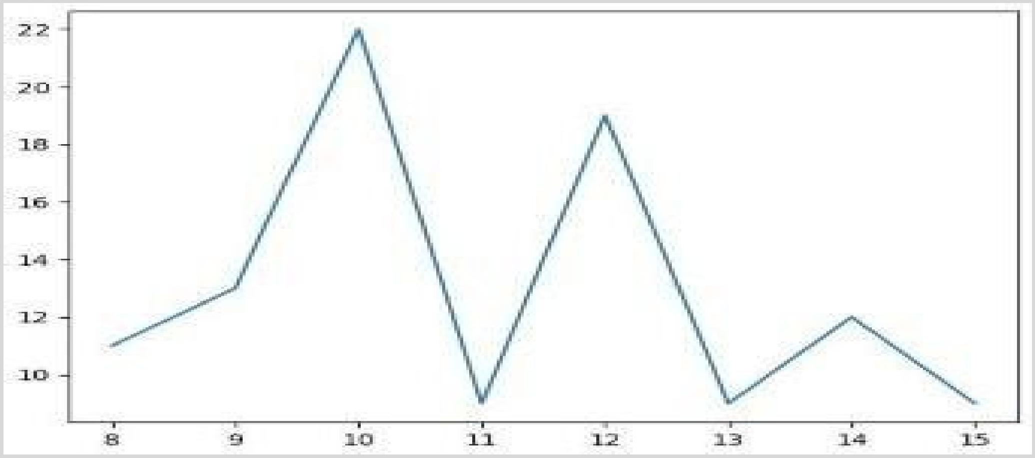
Methodology

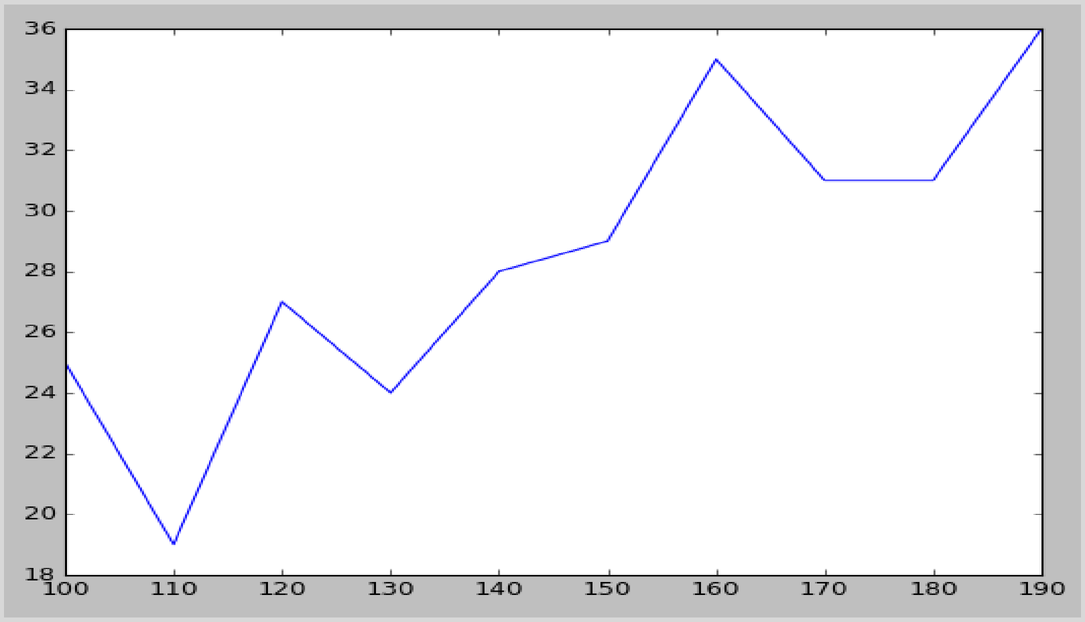
Standard methodology was followed that is the standard ACO algorithm for TSP, by initially keeping all ants randomly at some vertex.

Algorithm

```
For MAX_ITERATIONS:
            RESET ALL ANTS
                While all ants complete their tour:
             Choose next destination for each ant based on
          pheromone level Calculate the best paths
 Return best
paths Results
```

Various variations have been observed such as follows . Each of the paths were developed based on test cases generated by the code itself . The number of experiments/Iterations might vary each time . Here are some of them based on the different sets of co-ordinates chosen in every experiment for 2 different simulations .





Smith Waterman Algorithm:

The objective of smith waterman algorithm is to perform the local alignment of amino acid and nucleotide sequences. This alignment can be used to understand the structural and functional nature of them.

Steps of Smith Waterman Algorithm Include :-

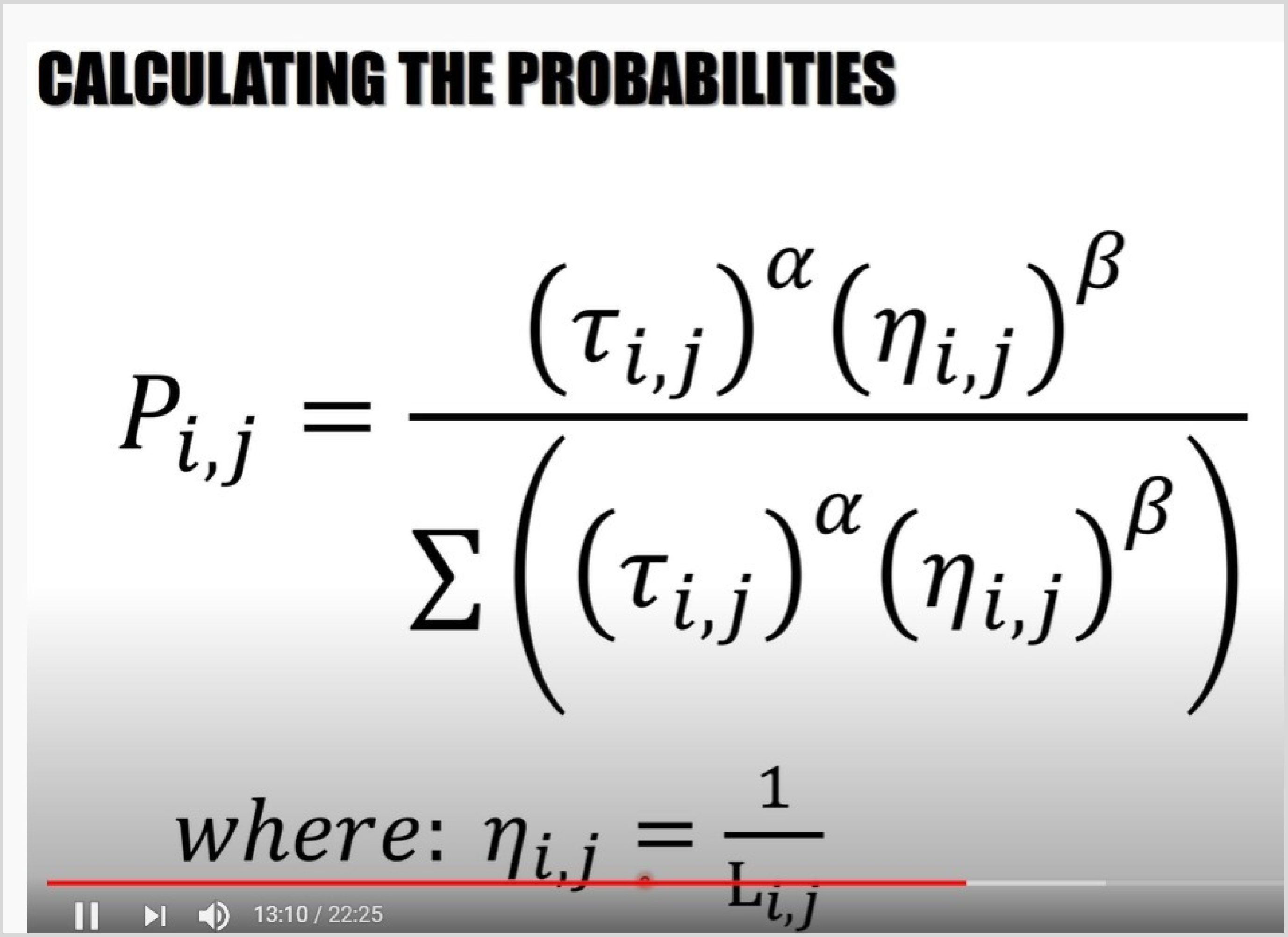
1. Initialization of a matrix.

2. Matrix Filling with the appropriate scores.

3. Trace back the sequences for a suitable alignment.

The formula for the matrix filling :-

$$M_{i,j} = Maximum\left[M_{i-1,j-1} + S_{i,j},\ M_{i,j-1} + W,\ M_{i-1,j} + W, 0\right]$$

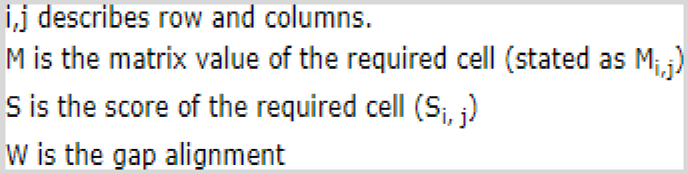


CALCULATING THE PROBABILITIES

$$P_{i,j} = \frac{(\tau_{i,j})^{\alpha}(\eta_{i,j})^{\beta}}{\sum\left((\tau_{i,j})^{\alpha}(\eta_{i,j})^{\beta}\right)}$$

$$\text{where: } \eta_{i,j} = \frac{1}{L_{i,j}}$$

Variables used



i,j describes row and columns.

M is the matrix value of the required cell (stated as $M_{i,j}$)

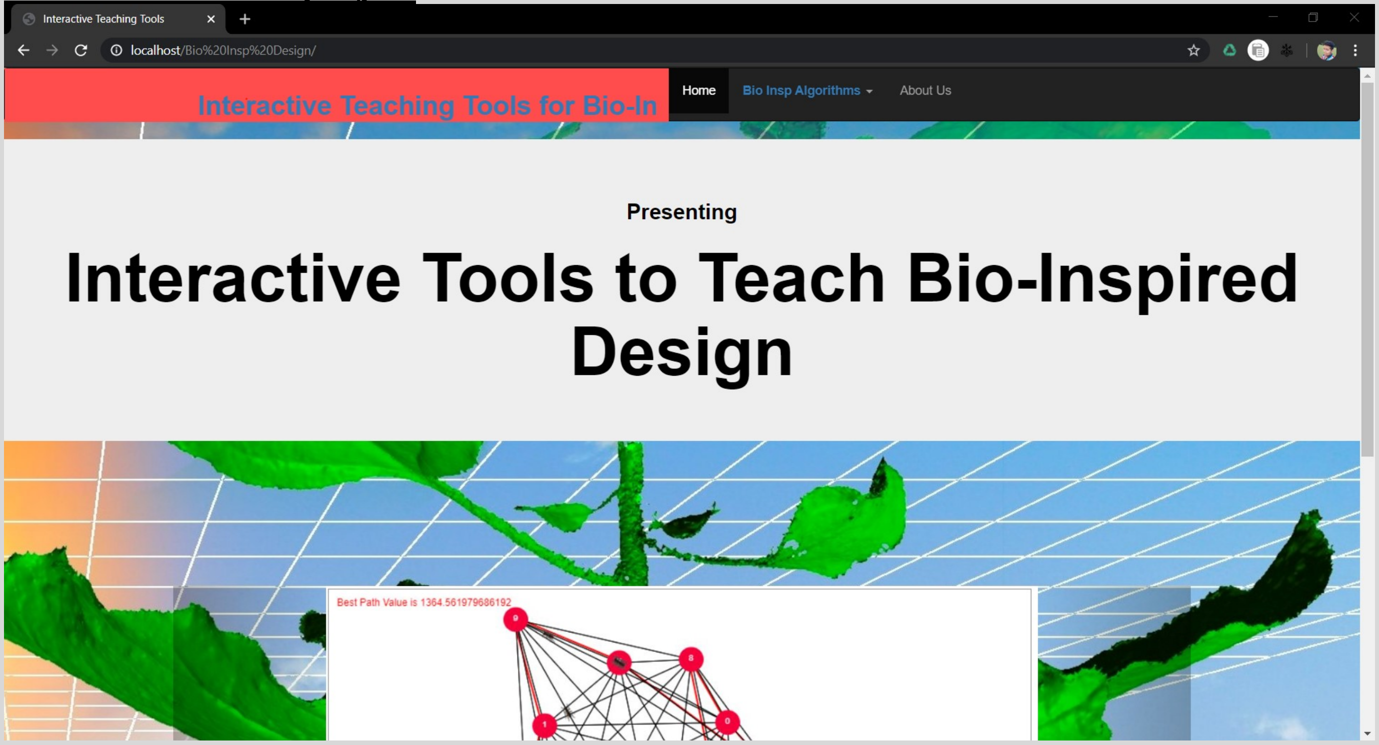
S is the score of the required cell ($S_{i,\ j}$)

W is the gap alignment

Result :-

The code follows the above equation to generate the local alignment of 2 given sequences

Screenshots of the project:


Interactive Teaching Tools
localhost/Bio%20Insp%20Design/
Interactive Teaching Tools for Bio-In
Home
Bio Insp Algorithms
About Us
Presenting
Interactive Tools to Teach Bio-Inspired Design
Best Path Value is 1364.561979686192

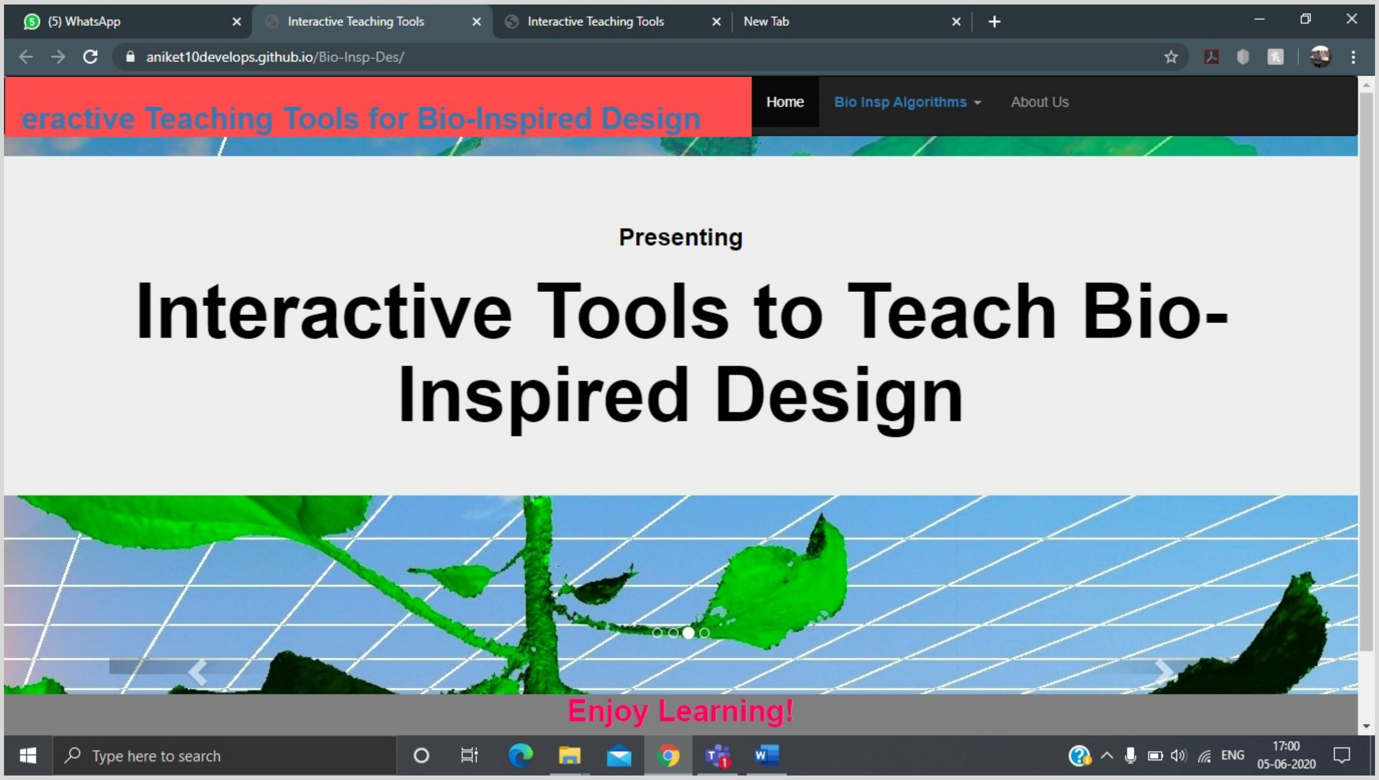
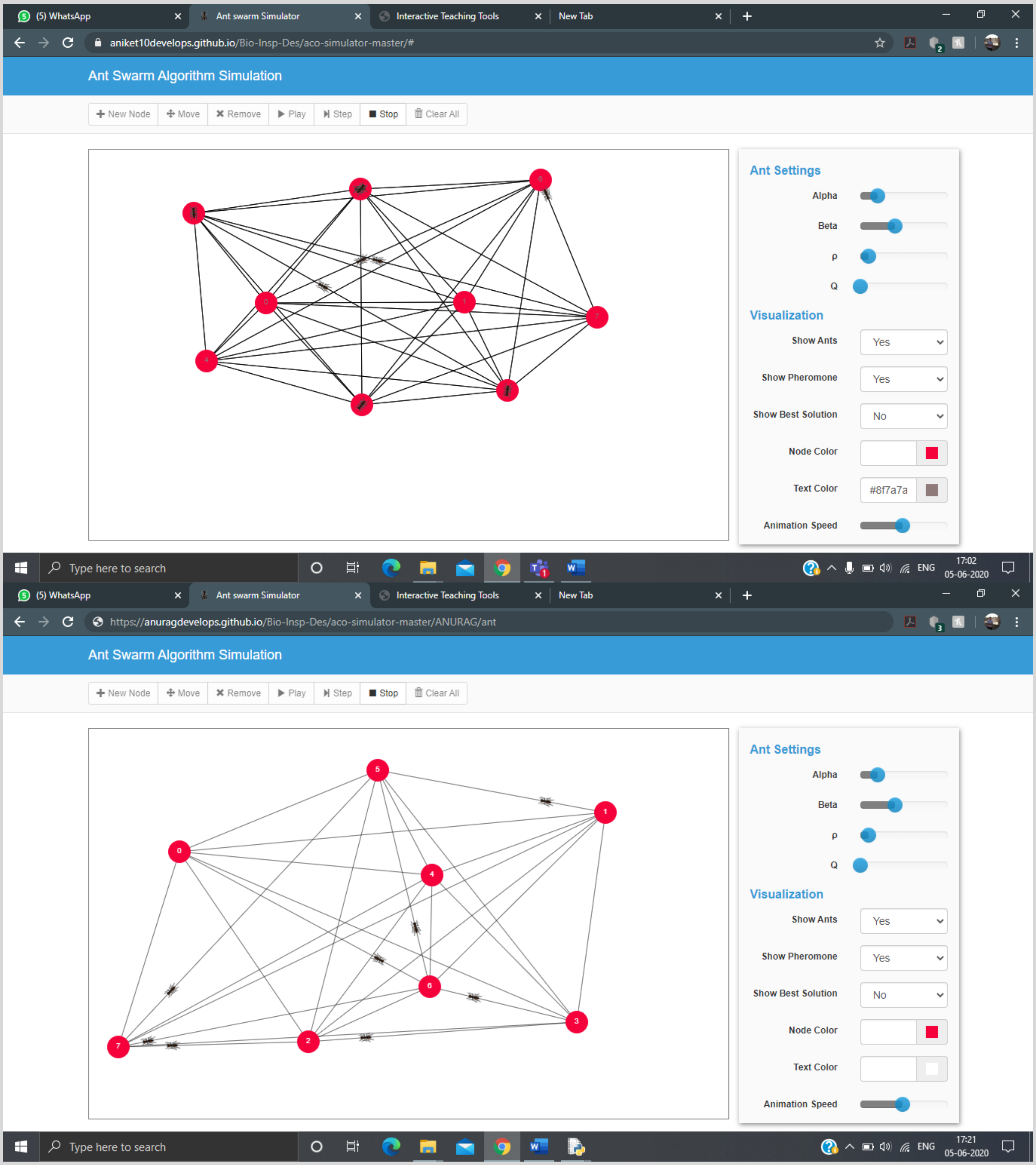

(5) WhatsApp
Interactive Teaching Tools
Interactive Teaching Tools
New Tab
aniket10develops.github.io/Bio-Insp-Des/
eractive Teaching Tools for Bio-Inspired Design
Home
Bio Insp Algorithms
About Us
Presenting
Interactive Tools to Teach Bio-Inspired Design
Enjoy Learning!
Type here to search
ENG
17:00
05-06-2020

Fig- opening web page of project.


(5) WhatsApp
Ant swarm Simulator
Interactive Teaching Tools
New Tab
aniket10develops.github.io/Bio-Insp-Des/aco-simulator-master/#
Ant Swarm Algorithm Simulation
New Node
Move
Remove
Play
Step
Stop
Clear All
Type here to search
ENG
17:02
05-06-2020

Ant Settings

Alpha

Beta

ρ

Q

Visualization

Show Ants: Yes

Show Pheromone: Yes

Show Best Solution: No

Node Color

Text Color: #8f7a7a

Animation Speed

(5) WhatsApp
Ant swarm Simulator
Interactive Teaching Tools
New Tab
https://anuragdevelops.github.io/Bio-Insp-Des/aco-simulator-master/ANURAG/ant
Ant Swarm Algorithm Simulation
New Node
Move
Remove
Play
Step
Stop
Clear All
5
1
0
4
6
3
7
2
Type here to search
ENG
17:21
05-06-2020

Ant Settings

Alpha

Beta

ρ

Q

Visualization

Show Ants: Yes

Show Pheromone: Yes

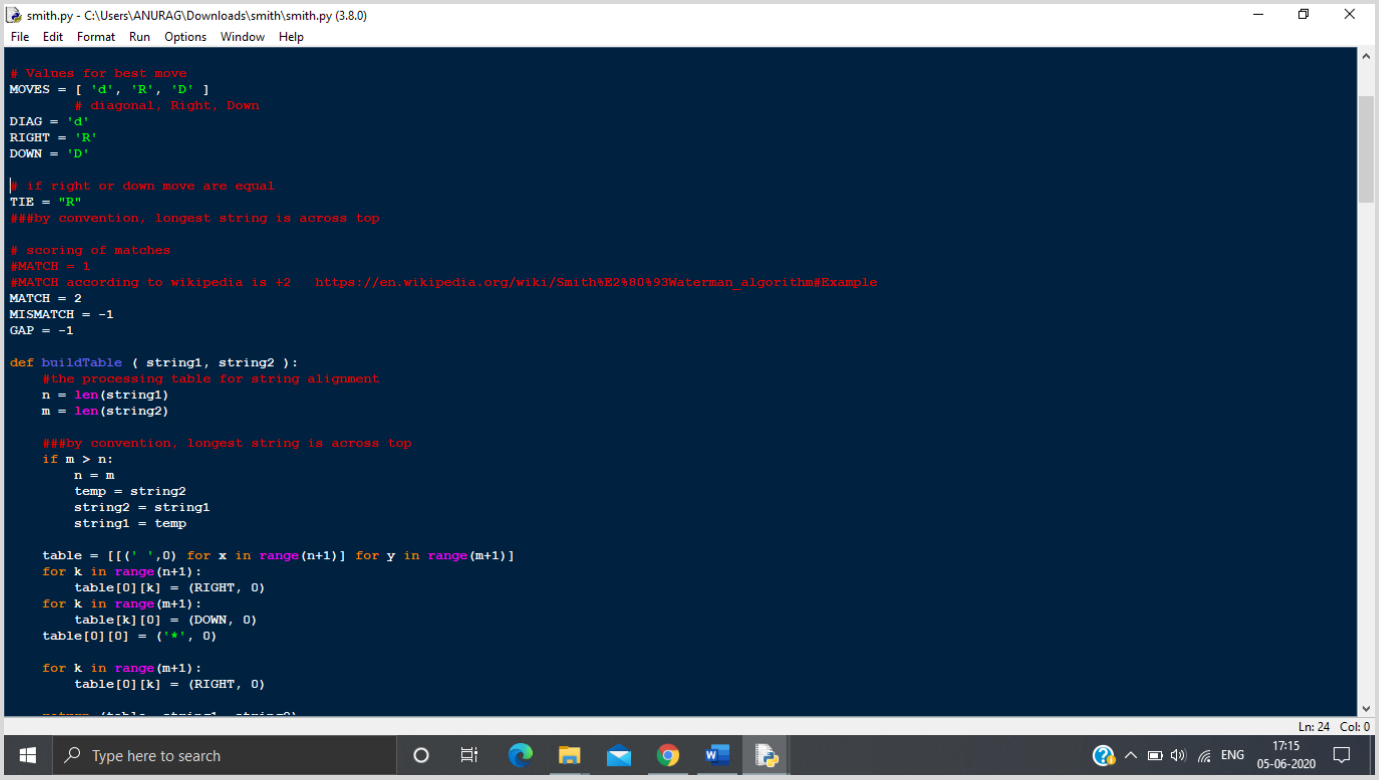
Show Best Solution: No

Node Color

Text Color

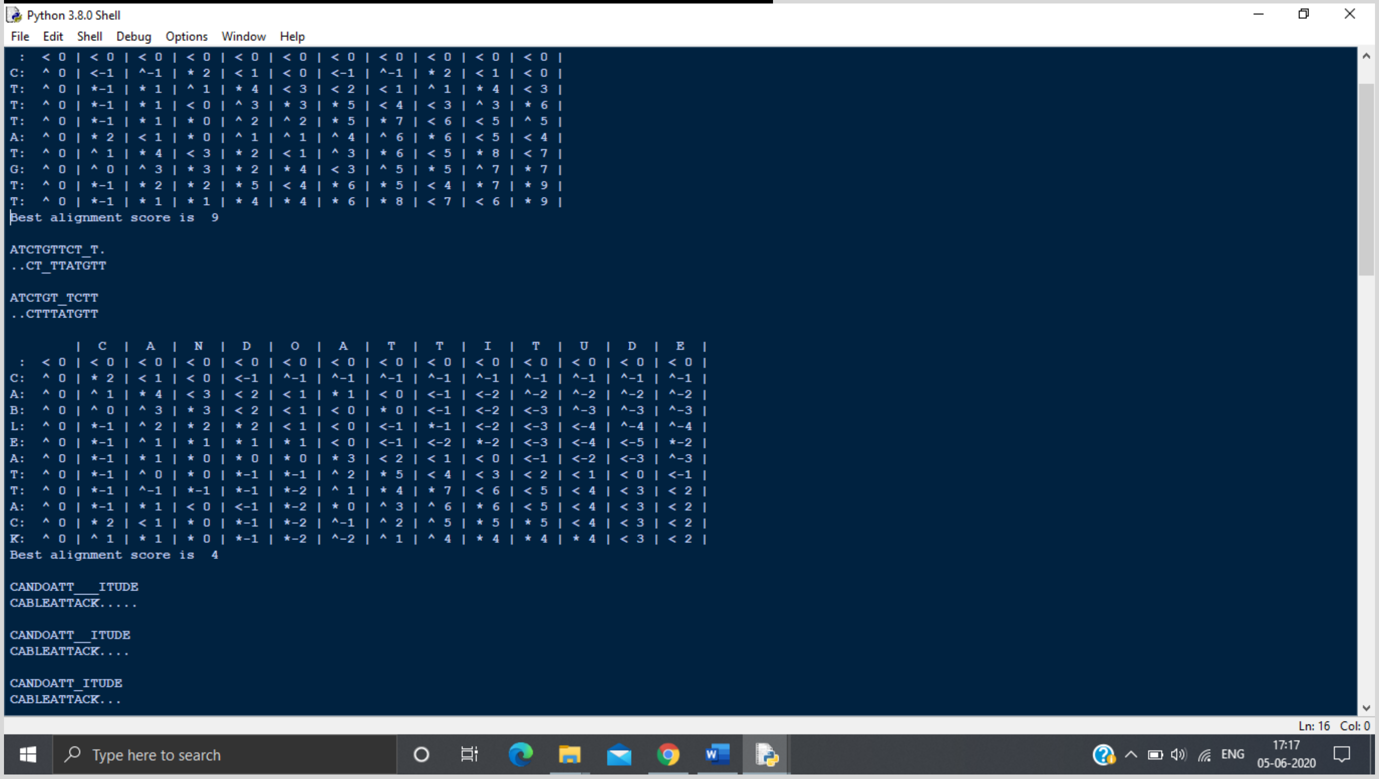
Animation Speed

<u>Fig- Window for ant swarm algorithm simulation</u>.

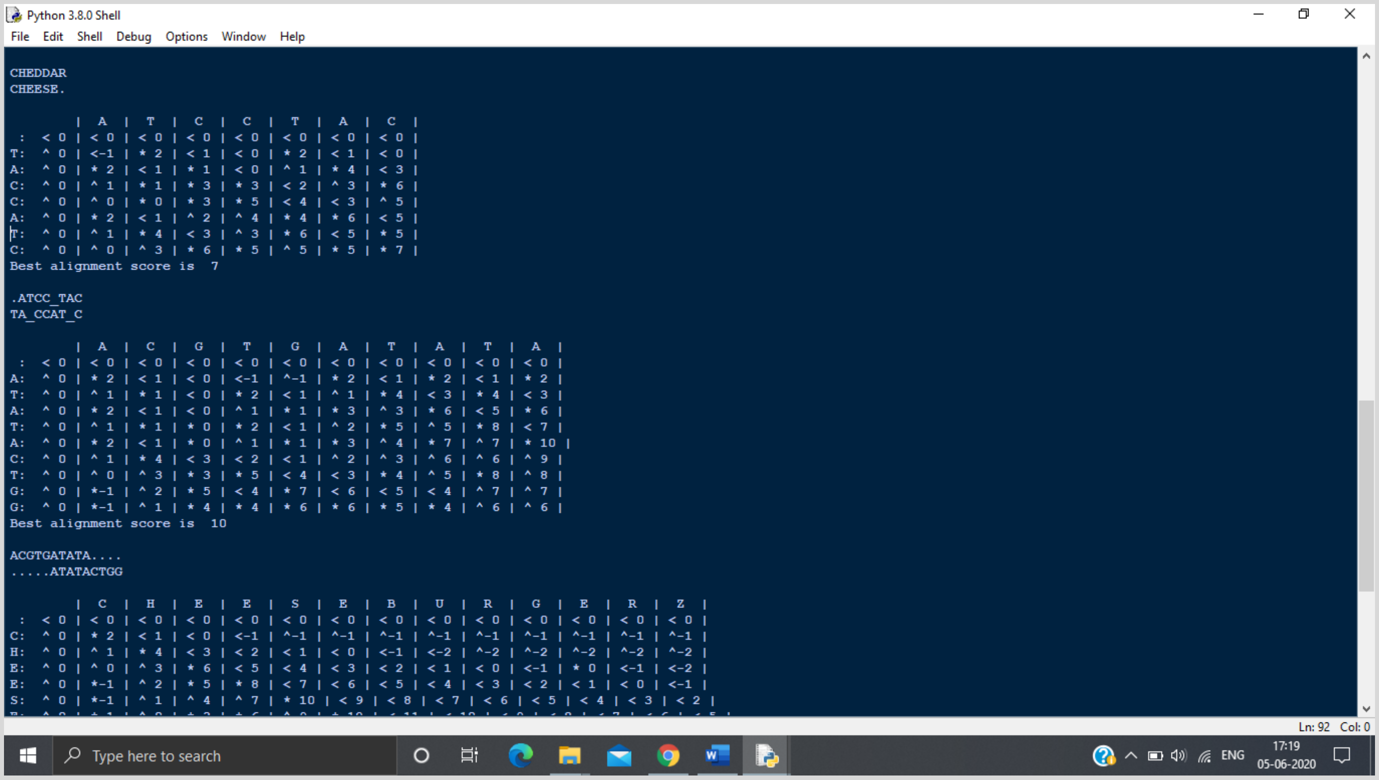


```
# Values for best move
MOVES = [ 'd', 'R', 'D' ]
        # diagonal, Right, Down
DIAG = 'd'
RIGHT = 'R'
DOWN = 'D'

# if right or down move are equal
TIE = "R"
###by convention, longest string is across top

# scoring of matches
#MATCH = 1
#MATCH according to wikipedia is +2   https://en.wikipedia.org/wiki/Smith%E2%80%93Waterman_algorithm#Example
MATCH = 2
MISMATCH = -1
GAP = -1

def buildTable ( string1, string2 ):
    #the processing table for string alignment
    n = len(string1)
    m = len(string2)

    ###by convention, longest string is across top
    if m > n:
        n = m
        temp = string2
        string2 = string1
        string1 = temp

    table = [[(' ',0) for x in range(n+1)] for y in range(m+1)]
    for k in range(n+1):
        table[0][k] = (RIGHT, 0)
    for k in range(m+1):
        table[k][0] = (DOWN, 0)
    table[0][0] = ('*', 0)

    for k in range(m+1):
        table[0][k] = (RIGHT, 0)
```

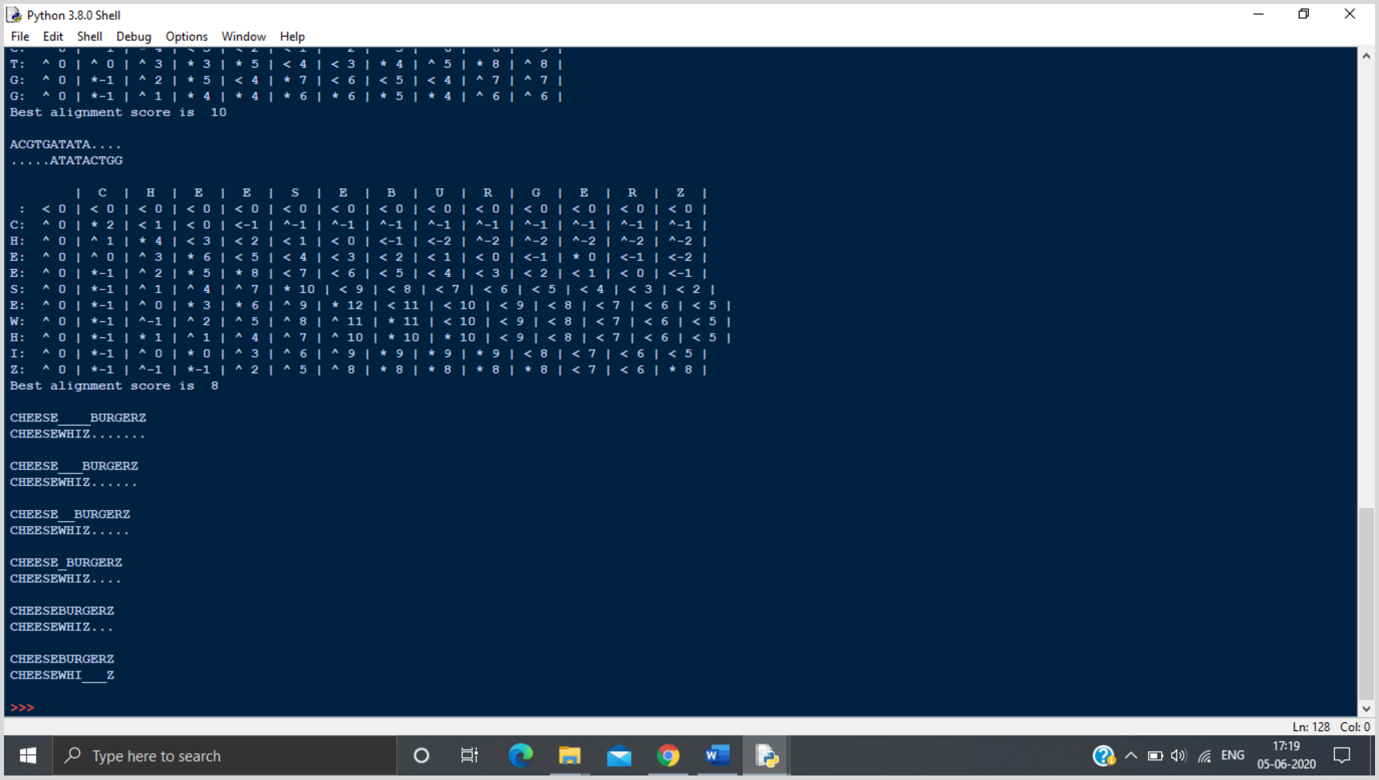
Fig- code for smith waterman on python console.



```
 :  < 0 | < 0 | < 0 | < 0 | < 0 | < 0 | < 0 | < 0 | < 0 | < 0 | < 0 |
C:  ^ 0 | <-1 | ^-1 | * 2 | < 1 | < 0 | <-1 | ^-1 | * 2 | < 1 | < 0 |
T:  ^ 0 | *-1 | * 1 | ^ 1 | * 4 | < 3 | < 2 | < 1 | ^ 1 | * 4 | < 3 |
T:  ^ 0 | *-1 | * 1 | < 0 | ^ 3 | * 3 | * 5 | < 4 | < 3 | ^ 3 | * 6 |
T:  ^ 0 | *-1 | * 1 | * 0 | ^ 2 | ^ 2 | * 5 | * 7 | < 6 | < 5 | ^ 5 |
A:  ^ 0 | * 2 | < 1 | * 0 | ^ 1 | ^ 1 | ^ 4 | ^ 6 | * 6 | < 5 | < 4 |
T:  ^ 0 | ^ 1 | * 4 | < 3 | * 2 | < 1 | ^ 3 | * 6 | < 5 | * 8 | < 7 |
G:  ^ 0 | ^ 0 | ^ 3 | * 3 | * 2 | * 4 | < 3 | ^ 5 | * 5 | ^ 7 | * 7 |
T:  ^ 0 | *-1 | * 2 | * 2 | * 5 | < 4 | * 6 | * 5 | < 4 | * 7 | * 9 |
T:  ^ 0 | *-1 | * 1 | * 1 | * 4 | * 4 | * 6 | * 8 | < 7 | < 6 | * 9 |
Best alignment score is  9

ATCTGTTCT_T.
..CT_TTATGTT

ATCTGT_TCTT
..CTTTATGTT

        |  C  |  A  |  N  |  D  |  O  |  A  |  T  |  T  |  I  |  T  |  U  |  D  |  E  |
 :  < 0 | < 0 | < 0 | < 0 | < 0 | < 0 | < 0 | < 0 | < 0 | < 0 | < 0 | < 0 | < 0 | < 0 |
C:  ^ 0 | * 2 | < 1 | < 0 | <-1 | ^-1 | ^-1 | ^-1 | ^-1 | ^-1 | ^-1 | ^-1 | ^-1 | ^-1 |
A:  ^ 0 | ^ 1 | * 4 | < 3 | < 2 | < 1 | * 1 | < 0 | <-1 | <-2 | ^-2 | ^-2 | ^-2 | ^-2 |
B:  ^ 0 | ^ 0 | ^ 3 | * 3 | < 2 | < 1 | < 0 | * 0 | <-1 | <-2 | <-3 | ^-3 | ^-3 | ^-3 |
L:  ^ 0 | *-1 | ^ 2 | * 2 | * 2 | < 1 | < 0 | <-1 | *-1 | <-2 | <-3 | <-4 | ^-4 | ^-4 |
E:  ^ 0 | *-1 | ^ 1 | * 1 | * 1 | * 1 | < 0 | <-1 | <-2 | *-2 | <-3 | <-4 | <-5 | *-2 |
A:  ^ 0 | *-1 | * 1 | * 0 | * 0 | * 0 | * 3 | < 2 | < 1 | < 0 | <-1 | <-2 | <-3 | ^-3 |
T:  ^ 0 | *-1 | ^ 0 | * 0 | *-1 | *-1 | ^ 2 | * 5 | < 4 | < 3 | < 2 | < 1 | < 0 | <-1 |
T:  ^ 0 | *-1 | ^-1 | *-1 | *-1 | *-2 | ^ 1 | * 4 | * 7 | < 6 | < 5 | < 4 | < 3 | < 2 |
A:  ^ 0 | *-1 | * 1 | < 0 | <-1 | *-2 | * 0 | ^ 3 | ^ 6 | * 6 | < 5 | < 4 | < 3 | < 2 |
C:  ^ 0 | * 2 | < 1 | * 0 | *-1 | *-2 | ^-1 | ^ 2 | ^ 5 | * 5 | * 5 | < 4 | < 3 | < 2 |
K:  ^ 0 | ^ 1 | * 1 | * 0 | *-1 | *-2 | ^-2 | ^ 1 | ^ 4 | * 4 | * 4 | * 4 | < 3 | < 2 |
Best alignment score is  4

CANDOATT___ITUDE
CABLEATTACK.....

CANDOATT__ITUDE
CABLEATTACK....

CANDOATT_ITUDE
CABLEATTACK...
```



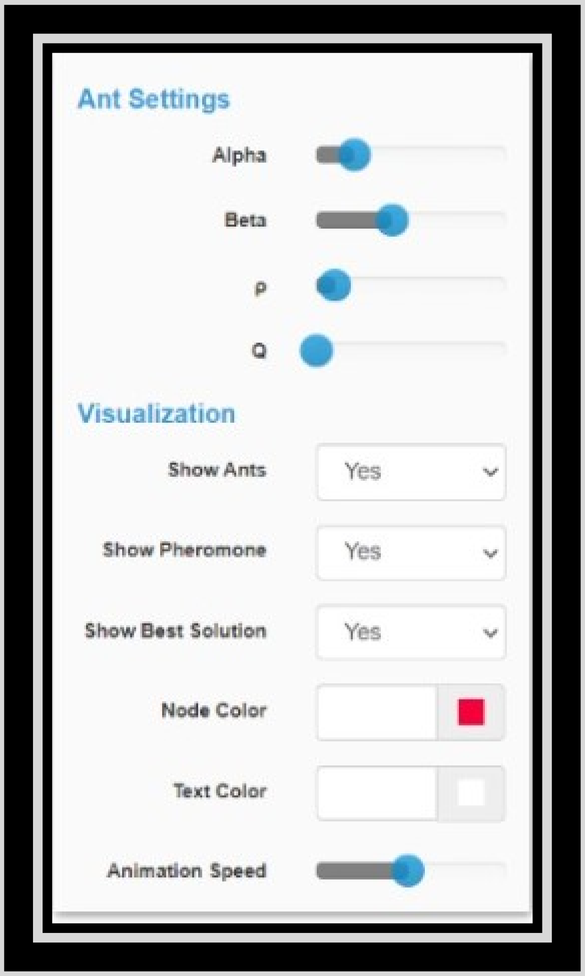
```
CHEDDAR
CHEESE.

         |  A  |  T  |  C  |  C  |  T  |  A  |  C  |
 :  < 0 | < 0 | < 0 | < 0 | < 0 | < 0 | < 0 | < 0 |
T:  ^ 0 | <-1 | * 2 | < 1 | < 0 | * 2 | < 1 | < 0 |
A:  ^ 0 | * 2 | < 1 | * 1 | < 0 | ^ 1 | * 4 | < 3 |
C:  ^ 0 | ^ 1 | * 1 | * 3 | * 3 | < 2 | ^ 3 | * 6 |
C:  ^ 0 | ^ 0 | * 0 | * 3 | * 5 | < 4 | < 3 | ^ 5 |
A:  ^ 0 | * 2 | < 1 | ^ 2 | ^ 4 | * 4 | * 6 | < 5 |
T:  ^ 0 | ^ 1 | * 4 | < 3 | ^ 3 | * 6 | < 5 | * 5 |
C:  ^ 0 | ^ 0 | ^ 3 | * 6 | * 5 | ^ 5 | * 5 | * 7 |
Best alignment score is  7

.ATCC_TAC
TA_CCAT_C

         |  A  |  C  |  G  |  T  |  G  |  A  |  T  |  A  |  T  |  A  |
 :  < 0 | < 0 | < 0 | < 0 | < 0 | < 0 | < 0 | < 0 | < 0 | < 0 | < 0 |
A:  ^ 0 | * 2 | < 1 | < 0 | <-1 | ^-1 | * 2 | < 1 | * 2 | < 1 | * 2 |
T:  ^ 0 | ^ 1 | * 1 | < 0 | * 2 | < 1 | ^ 1 | * 4 | < 3 | * 4 | < 3 |
A:  ^ 0 | * 2 | < 1 | < 0 | ^ 1 | * 1 | * 3 | ^ 3 | * 6 | < 5 | * 6 |
T:  ^ 0 | ^ 1 | * 1 | * 0 | * 2 | < 1 | ^ 2 | * 5 | ^ 5 | * 8 | < 7 |
A:  ^ 0 | * 2 | < 1 | * 0 | ^ 1 | * 1 | * 3 | ^ 4 | * 7 | ^ 7 | * 10 |
C:  ^ 0 | ^ 1 | * 4 | < 3 | < 2 | < 1 | ^ 2 | ^ 3 | ^ 6 | ^ 6 | ^ 9 |
T:  ^ 0 | ^ 0 | ^ 3 | * 3 | * 5 | < 4 | < 3 | * 4 | ^ 5 | * 8 | ^ 8 |
G:  ^ 0 | *-1 | ^ 2 | * 5 | < 4 | * 7 | < 6 | < 5 | < 4 | ^ 7 | ^ 7 |
G:  ^ 0 | *-1 | ^ 1 | * 4 | * 4 | * 6 | * 6 | * 5 | * 4 | ^ 6 | ^ 6 |
Best alignment score is  10

ACGTGATATA....
.....ATATACTGG

         |  C  |  H  |  E  |  E  |  S  |  E  |  B  |  U  |  R  |  G  |  E  |  R  |  Z  |
 :  < 0 | < 0 | < 0 | < 0 | < 0 | < 0 | < 0 | < 0 | < 0 | < 0 | < 0 | < 0 | < 0 | < 0 |
C:  ^ 0 | * 2 | < 1 | < 0 | <-1 | ^-1 | ^-1 | ^-1 | ^-1 | ^-1 | ^-1 | ^-1 | ^-1 | ^-1 |
H:  ^ 0 | ^ 1 | * 4 | < 3 | < 2 | < 1 | < 0 | <-1 | <-2 | ^-2 | ^-2 | ^-2 | ^-2 | ^-2 |
E:  ^ 0 | ^ 0 | ^ 3 | * 6 | < 5 | < 4 | < 3 | < 2 | < 1 | < 0 | <-1 | * 0 | <-1 | <-2 |
E:  ^ 0 | *-1 | ^ 2 | * 5 | * 8 | < 7 | < 6 | < 5 | < 4 | < 3 | < 2 | < 1 | < 0 | <-1 |
S:  ^ 0 | *-1 | ^ 1 | ^ 4 | ^ 7 | * 10 | < 9 | < 8 | < 7 | < 6 | < 5 | < 4 | < 3 | < 2 |
```



```
T:  ^ 0 | ^ 0 | ^ 3 | * 3 | * 5 | < 4 | < 3 | * 4 | ^ 5 | * 8 | ^ 8 |
G:  ^ 0 | *-1 | ^ 2 | * 5 | < 4 | * 7 | < 6 | < 5 | < 4 | ^ 7 | ^ 7 |
G:  ^ 0 | *-1 | ^ 1 | * 4 | * 4 | * 6 | * 6 | * 5 | * 4 | ^ 6 | ^ 6 |
Best alignment score is  10

ACGTGATATA....
.....ATATACTGG

         |  C  |  H  |  E  |  E  |  S  |  E  |  B  |  U  |  R  |  G  |  E  |  R  |  Z  |
 :  < 0 | < 0 | < 0 | < 0 | < 0 | < 0 | < 0 | < 0 | < 0 | < 0 | < 0 | < 0 | < 0 | < 0 |
C:  ^ 0 | * 2 | < 1 | < 0 | <-1 | ^-1 | ^-1 | ^-1 | ^-1 | ^-1 | ^-1 | ^-1 | ^-1 | ^-1 |
H:  ^ 0 | ^ 1 | * 4 | < 3 | < 2 | < 1 | < 0 | <-1 | <-2 | ^-2 | ^-2 | ^-2 | ^-2 | ^-2 |
E:  ^ 0 | ^ 0 | ^ 3 | * 6 | < 5 | < 4 | < 3 | < 2 | < 1 | < 0 | <-1 | * 0 | <-1 | <-2 |
E:  ^ 0 | *-1 | ^ 2 | * 5 | * 8 | < 7 | < 6 | < 5 | < 4 | < 3 | < 2 | < 1 | < 0 | <-1 |
S:  ^ 0 | *-1 | ^ 1 | ^ 4 | ^ 7 | * 10 | < 9 | < 8 | < 7 | < 6 | < 5 | < 4 | < 3 | < 2 |
E:  ^ 0 | *-1 | ^ 0 | * 3 | * 6 | ^ 9 | * 12 | < 11 | < 10 | < 9 | < 8 | < 7 | < 6 | < 5 |
W:  ^ 0 | *-1 | ^-1 | ^ 2 | ^ 5 | ^ 8 | ^ 11 | * 11 | < 10 | < 9 | < 8 | < 7 | < 6 | < 5 |
H:  ^ 0 | *-1 | * 1 | ^ 1 | ^ 4 | ^ 7 | ^ 10 | * 10 | * 10 | < 9 | < 8 | < 7 | < 6 | < 5 |
I:  ^ 0 | *-1 | ^ 0 | * 0 | ^ 3 | ^ 6 | ^ 9 | * 9 | * 9 | * 9 | < 8 | < 7 | < 6 | < 5 |
Z:  ^ 0 | *-1 | ^-1 | *-1 | ^ 2 | ^ 5 | ^ 8 | * 8 | * 8 | * 8 | * 8 | < 7 | < 6 | * 8 |
Best alignment score is  8

CHEESE____BURGERZ
CHEESEWHIZ.......

CHEESE___BURGERZ
CHEESEWHIZ......

CHEESE__BURGERZ
CHEESEWHIZ.....

CHEESE_BURGERZ
CHEESEWHIZ....

CHEESEBURGERZ
CHEESEWHIZ...

CHEESEBURGERZ
CHEESEWHI___Z

>>>
```

Fig- output for smith waterman algorithm on python shell(3.8.0)

<u>Palette for setting of Ant colony Simulation:</u>


Ant Settings

Alpha

Beta

P

Q

Visualization

Show Ants: Yes

Show Pheromone: Yes

Show Best Solution: Yes

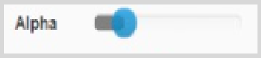
Node Color

Text Color

Animation Speed

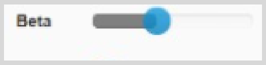
<u>Functions of sliders Under ant settings</u>

<u>ALPHA (α)</u>

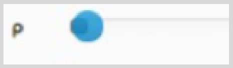


This parameter helps to control the amount of pheromones secreted by the ants While finding their paths.

<u>Beta (β)</u>



This parameter helps to control the attraction of other possible transition from One node to another.

Rho(ρ) 

This parameter decides the pheromone evaporation rate from the path.

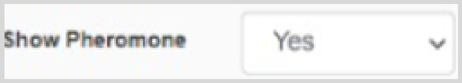
Q

This is a random variable uniformly distributes over[0,1] and a parameter q0.

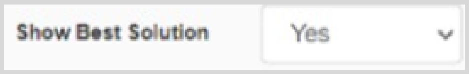
Functions under visualization

Show ants
Show Ants
Yes

This drop-down box help us to choose the option for visibility of ants on nodes
Or not.

Show pheromone 
Show Pheromone
Yes

This options help us to choose to make pheromones visible on nodes or not.

Show Best Solution 
Show Best Solution
Yes

This option helps user to show the shortest path covered by ants over the selected
Nodes.

Node colour
Node Color

This option helps user to choose the color of nodes.

Text colour
Text Color

This option maintains the text color display on the screen during simulation.

<u>Animation speed</u>

This option control the speed of ants walking on the proposed nodes for simulation.

REFERENCES

This topic is a rather fresh topic so the amount of literature and references available is limited**. For the Algorithms :-**

- Introduction to Genetic Algorithms – By Vijini Mallawaarachchi (accessed on 6th December 2018).

For the interactive web design :-

- Victor J Stevens, PhD[1], Kristine L Funk, MS, RD[1], Phillips J Bartley PhD[2] , Thomas P Eelinger ,MD,MHP[3] 'Design and Implementation of an Interactive Website to Support Long-Term Maintenance of Weight Loss' J Med International Res.2008 JanMar;10

Thanks!