WhiteSpace

WHITESPACE OPERATIONAL CODING CHALLENGES
GAME RULES

INTRO

Thank you for your interest in our coding challenges. We hope you enjoy them!

GUIDELINES

Before jumping in, please read through the following guidelines:

- We expect you to complete all tasks using Python 3.x.
- You can use any external resources/packages you want, but understanding the results is your own responsibility.
- You are not allowed to share (any element of) these challenges with third parties, including online fora such as GitHub.
- Wherever the challenge mentions the word `print` we expect a print statement in your code with the specified output.
- Remember to include helpful comments and docstrings in your code. State any assumptions you make in your comments.

DATA

The assignments contain references to data sets. You can find these sets here:
https://tinyurl.com/92zpsv7v

CODE ORGANIZATION

- Please organize your code so your answer to each challenge is stored in a separate file.
- Please include a main method and an `if __name__ == '__main__'` block in each file. This block should be used to run the main method.
- Please store your code and data in the same folder.

DELIVERY

- Each file should be called `{your_name}_{challenge_name}.{extension}` with the extension being either .py or .ipynb
- Please combine all files in a single .zip called `{your_name}`*`_whitespace_codingchallenges_`*`{YYYYMMDD}.zip`
- Please e-mail your zipped results to careers@whitespacesolutions.eu
- You are expected to send your results within 3 hours of receiving this email. Late submissions will not be considered.

Enjoy! – The WhiteSpace team

WHITE SPACE CODING CHALLENGE 1

NUANCED NAVIGATION

INTRODUCTION

You are part of a sailing crew and are responsible for the route planning of an upcoming trip.

Your briefing:

- You have been given a list of 12 crucial sites to visit. Your mission is to design an optimal path that connects these sites. The path needs to be cyclic: it can start at any site but can only visit each subsequent site once.
- The sailing vessel has one constraint: it cannot sail directly against the wind. Since the wind in the area comes from the North, the vessel cannot navigate directly South → North. Any other direction is fine. (On the map below, this means for example that the vessel can sail from 6 to 4, but not from 4 to 6.)
- Your task is to plan for optimality: the shorter the total distance travelled, the better.

ASSIGNMENTS

- Write a Python function that finds the cycle that connects all sites and minimizes total distance travelled. Remember, each site should be visited only once, and the vessel cannot directly travel South → North.
- `print` the *order* in which the cycle visits the path, and the *total distance* travelled.

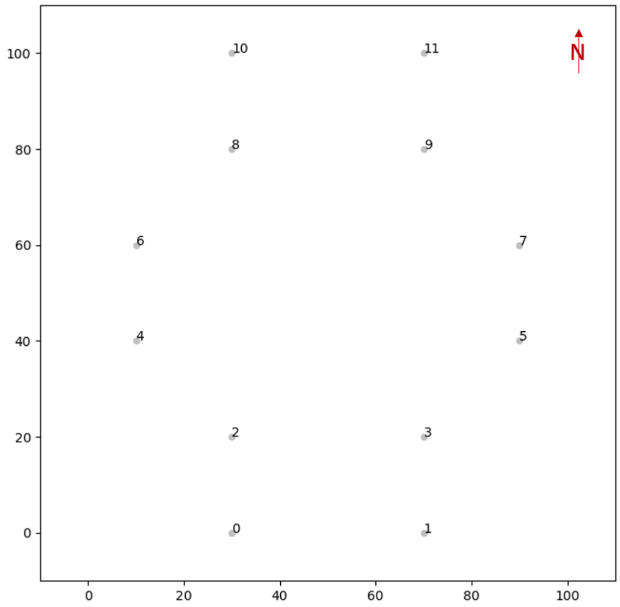

Figure: illustration of the Nuanced Navigation data. Each dot represents a site, labelled 0-11.

WhiteSpace

WHITE SPACE CODING CHALLENGE 2

TURBULENT TURBINES

INTRODUCTION

You've been appointed as a data analyst at a green energy company. Your first assignment is to analyze the performance of a wind farm, composed of several types of wind turbines. You have been given two datasets:

- a CSV file containing time series data for each turbine (turbine id, power output and timestamp).
- a CSV file containing metadata for each turbine (turbine id and turbine type).

The data you received is imperfect: certain records have a null power output entry, indicating a temporary turbine failure.

ASSIGNMENTS

- Write a Python function that reads in these two CSV files and combines them into a single dataset.
- Next, determine which turbine type had the lowest average power output in March, considering only non-null values. `print` the associated *model type* and its *average power output*.
- Finally, calculate the failure ratio (number of failures / number of readings) per turbine type. Find the turbine type with the highest failure ratio. `print` its *type* and its *failure ratio*.

WhiteSpace

WHITE SPACE CODING CHALLENGE 3

WILD WAVES

INTRODUCTION

You are an avid surfer and are planning your surfing activities along the coast. You have access to two time series:

- a CSV file with wave height (in meters) versus time (in hours).
- a CSV file with the wave height threshold (in meters) versus time (in hours). This threshold describes the maximum allowed wave height for safe surfing.

ASSIGNMENTS

- First, interpolate both time series to a common time grid.
- Next, determine the optimal surfing window, i.e., the longest uninterrupted window during which the wave height is at least 1 meter but below the threshold.
- `print` the *starting time* and *duration* of the optimal surfing window, rounded to the closest integer.
- Create a plot to visualize the time series (wave height and threshold), annotate the periods when the waves exceed the threshold, and highlight the optimal surfing window.

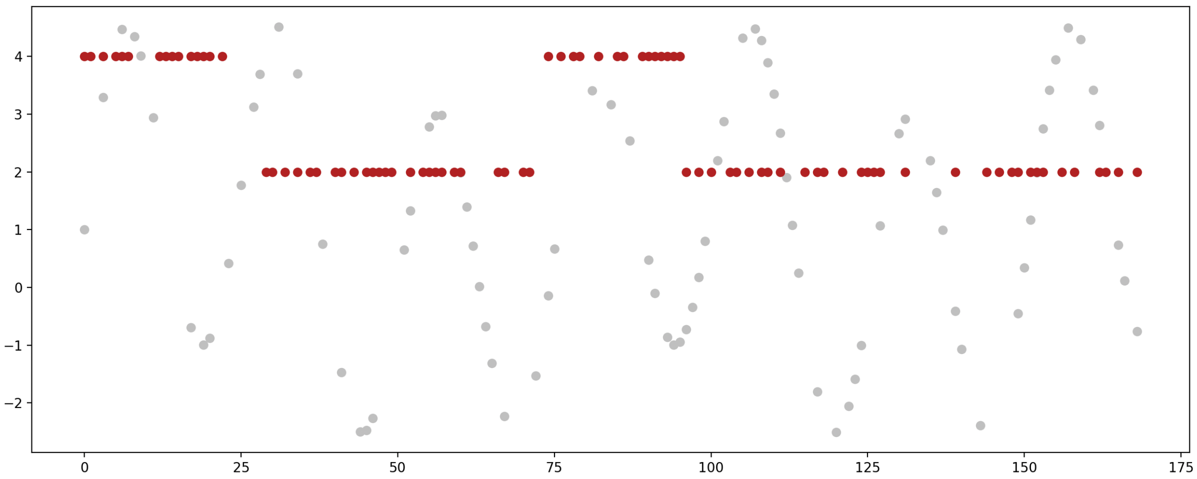

Figure: illustration of the Wild Waves data. The wave height is shown in grey, the threshold in red.

WHITE SPACE CODING CHALLENGE 4

GENOME GEMS

INTRODUCTION

Imagine you're a bioinformatics researcher studying the DNA sequences of different species. You're particularly interested in finding commonalities in these sequences, as these could indicate shared biological traits or ancestry.

The data you've received represents parts of DNA sequences from different species. Each sequence is represented as a string of characters: A, T, C, G.

You are tasked with finding the longest common substring in the DNA sequences of all these species. A substring is a contiguous sequence of characters within the DNA sequence. For example, in the string 'ACGT', 'CG' is a substring. Substrings must maintain the order of characters as they appear in the original string, without any intervening characters.

To account for slight variations, you are allowed to consider a common substring that has at most one mutation (difference) across the species (see Figure below). Note that the position of the mutation may differ across species.

ASSIGNMENTS

- First, write a Python function that reads in the file and stores the DNA sequences in an appropriate data structure.
- Next, identify the longest common substring with at most one mutation per species.
- `print` *the longest common substring with at most one mutation* and its *length*.

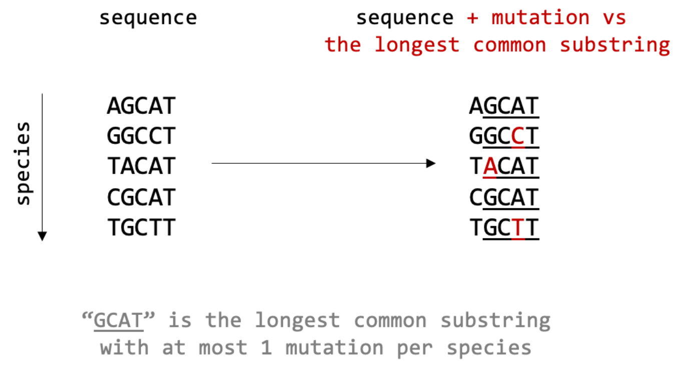
sequence
sequence + mutation vs
the longest common substring
species
AGCAT
GGCCT
TACAT
CGCAT
TGCTT
AGCAT
GGCCT
TACAT
CGCAT
TGCTT
"GCAT" is the longest common substring
with at most 1 mutation per species

Figure: illustration of the Genome Gems data.

WhiteSpace

WHITE SPACE CODING CHALLENGE 5

DRONE DILEMMAS

INTRODUCTION

You are a drone operations analyst, responsible for the safety of a drone delivery system across a bustling city. Your task is to identify potential mid-air drone collisions to improve the safety of the system.

You are provided with a file containing the following data for each drone:

- id
- start position
- end position
- start time (in minutes)
- speed (meters per minute)

ASSIGNMENTS

- First, write a Python function that reads in the provided data and stores it in an appropriate data structure.
- Next, create a function that identifies drones "at risk". Drones are considered at risk if, at any time, they come within 100 meters of any other drone.
- Finally, `print` the ids of the drones that are at risk.

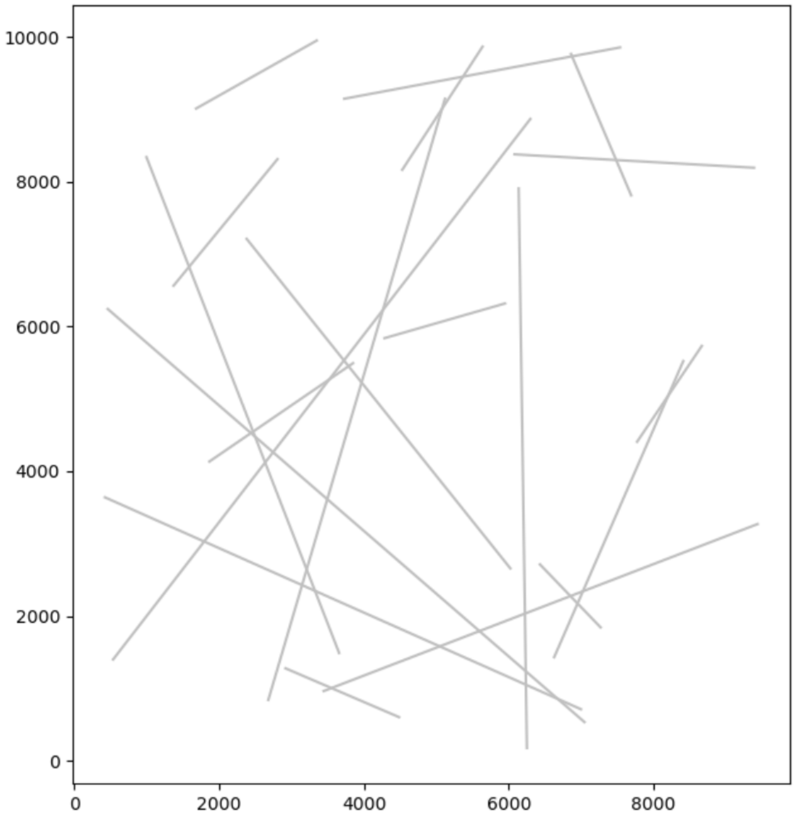

Figure: illustration of the Drone Dilemmas data. Each line represents an aerial view of a drone path.

WHITE SPACE CODING CHALLENGES

POTENTIALLY RELEVANT LINKS

- https://developers.google.com/optimization
- https://matplotlib.org/stable/index.html
- https://numpy.org/doc/
- https://pandas.pydata.org/docs/
- https://pypi.org/project/utm/
- https://scipy.org/
- https://shapely.readthedocs.io/en/stable/manual.html