FIT1045/53 Algorithmic Problem Solving – Workshop 4.

Objectives

The **objectives of this workshop** are:

- To implement and manipulate data structures for graphs in Python.
- To implement algorithms on graphs in Python.
- To implement tables as lists of lists in Python.

Useful Material

Reading from files: Section 7.2.1
`https://docs.python.org/3/tutorial/inputoutput.html#reading-and-writing-files`

Lists: `https://docs.python.org/3/tutorial/datastructures.html`

Task 0: Prelab

NOTE: If using Windows, it is encouraged that you open any text files being used from input/output in either the IDLE editor that comes standard with any Python installation. Or use a text editor that is better suited than Notepad. A good example is Notepad++.

The purpose of this prelab is to raise your understanding of reading/writing from/to files.

Part 1 The file 'mod_haiku.txt' contains a haiku poem with the spaces replaced with ';' and newlines, '\n', replaced with tabs, '\t'. Using only the following functions, and any control structures or datatypes you have learned thus far, you should restore the haiku to its previous state and write it to a file called 'haiku.txt':

```
open()
write()
close()
split()
join()
```

Part 2 The file 'grades.txt' constains a list of (fictional) student grades for FIT1045. The file contains several lines, where each line represents a unique student. The comma separated values are in the following order: student ID, full name and final mark. You are to write a program that reads the file and displays each field to the user with final grade (N, P, C, D, HD) calculated and displayed.

Task 1:

Part A

A palindrome is a word that is spelt the same forward and backwards. Your task is to write a function that takes, as input, a single word and returns `True` if it is a palindrome and `False` otherwise.

Part B

Using your function from Part A, write a program which reads words from a file. For each word, output "<word> is a palindrome" or if the word is in fact a palindrome, or "<word> is not a palindrome" if not. You can test you program on the file "palindromic"

Note: For FIT1045 students, for full marks, only one of tasks 2/3 needs to be completed (in addition to the other tasks). FIT1053 students need to complete all tasks for full marks.

Task 2:

For this task we will have you working with tidal data from `https://www.tide-forecast.com` for Melbourne, Australia. Have a look at the file `Tides.txt`, you'll see it follows the format (**date,hours_since_midnight,tide_in_meters**), for instance:

```
Friday 2 February,23.52,0.04 meters
...
Sunday 14 January,12.55,0.70 meters
...
etc.
```

Each day will typically have three or four measurements and the data is ordered by day and time.

Part A

Adapt your code from earlier tasks to read data from `Tides.txt` into the format:

```
['Friday 2 February',23.52,0.04]
...
['Sunday 14 January',12.55,0.70]
...
```

In this case you have a table, where each inner list is a list of [date,hours since midnight, meters] with the formats [string,float,float]

Part B

For each day, find out the lowest and highest tide in meters and print these to the screen for each day; for instance:

```
...
Wednesday 17 January: 0.06 meters at lowest and 0.70 meters at highest
Thursday 18 January: 0.05 meters at lowest and 0.73 meters at highest
...
```

Hint: Don't forget that the data is ordered by date, this means that all the values for Wednesday 17 January are right after each other

Part C

Prepare another table to hold the time (in hours since midnight) that the highest and lowest tide occured at as well as the corresponding tide in meters. For instance:

```
['Wednesday 17 January',22.37,0.06,4.52,0.70]
['Thursday 18 January',23.08,0.05,5.23,0.73]
```

Compute the average time (in hours since midnight) that the highest and lowest tides occur at over the full month of data values. For instance:

```
Over the full period, on average the lowest and highest tides
occurred at XX.XX and YY.YY hours after midnight
```

Task 3 - Mazes

In this task you will be investigating how to generate mazes using graphs and spanning trees.

Part A

Implement the spanning tree algorithm from the lectures. That is, write a function `spanning_tree(graph)` which takes as input a graph (in the form of an adjacency matrix) and outputs an adjacency matrix of a spanning tree of that graph. Note that you will need to implement the `empty_graph(n)` function. This function takes an input `n` and returns a `n`×`n` table full of 0s, representing the adjacency matrix of an empty graph with `n` vertices.

Part B

Implement a function `grid_graph(m, n)` that return an adjacency matrix of a complete grid graph with m rows and n columns. A complete grid graph is a graph with the vertices arranged in a grid, with each vertex connected to its horizontal and vertical neighbours. Note that for this task, although the graph is shaped like an `n`×`m` grid, the adjacency matrix will not be an `n`×`m` matrix. Think carefully about the relationship between the graph and the adjacency matrix representation.

Part C

Implement a function `print_grid(n, adjacency_matrix)` that prints a grid graph (using * for vertices, '---' and '|' for vertical and horizontal edges). `n` is the number of columns in the grid. Assume that the vertices in the adjacency matrix are the vertices in the grid in the order left-to-right, top-to-bottom.

Part D

Implement a function `maze(m, n)` which combines the functions written in the previous parts in order to generate a `m` × `n` maze. You may find that these mazes are quite boring. Use your knowledge of the `random` module to make them more interesting!

Task 4 - Optional (Not assessed)

If you have some time left over, consider attempting the following (very good experience working with files, tables and data sets)

Part A

Setup code which will write your table of tides and times to its own file. Adapt your program such that it converts hours since midnight into a time in 12 hour time (eg. 2:47 PM or 11:19 AM) so that your output is easier to read.

Part B

Navigate to the website: `https://www.tide-forecast.com/locations/Melbourne-Australia/tides/latest` and copy the table there into a .csv file. Prepare a program that can read the data in as a file (splitting around commas will give you each cell in a given line as CSVs are comma seperated) and convert it into the format used in **Task 3 Part A)**

Part C

Try and answer one of the following questions:

1. How are the tides affected by Month (you'll want to find more than just two months worth of data to answer this)
2. How are the tides affected by location (you'll want to consider places with significantly different latitudes and longitudes)

You can think about both in terms of meters and or times of highest and lowest tide.