

H2 Computing Practical Worksheet – T1W4

- 1a Write the code that generates 50 numeric values between 0 and 100, to 1 decimal place (following a uniform distribution) and saves those values to the text file `SCORES.TXT`.
- Note that each line of the file should store a single score.
- 1b Write the code to read the text file `SCORES.TXT` and output its contents in the form of a Stem and Leaf diagram.
- Details on the Stem and Leaf diagram can be found on the following pages:
<http://mathworld.wolfram.com/Stem-and-LeafDiagram.html>
<http://www.purplemath.com/modules/stemleaf.htm>
 - You are to ensure that your output corresponds to a valid Stem and Leaf diagram and that suitably representative of the stored data (refer to the notes above for details on this).
 - You are to ensure that you follow good user interface practices when designing your output.
 - You should not use any non-standard libraries such as numpy, scipy, matplotlib.
- 1c Write code to read the text file `SCORES.TXT` and output its contents in the form of a Histogram.
- Details on Histograms can be found on the following pages:
<http://mathworld.wolfram.com/Histogram.html>
<https://www.mathsisfun.com/data/histograms.html>
 - You are to ensure that your output corresponds to a valid Histogram and that suitably representative of the stored data (refer to the notes above for details on this).
 - You are to ensure that you follow good user interface practices when designing your output.
 - You should not use any non-standard libraries such as numpy, scipy, matplotlib.

- 1d Assume that you are to store the score information for multiple subjects. Initially you are told to store these in a multi-dimensional array, such that it follows the form:

```
scores[student_id][sub1_score][sub2_score]...[subn_score]
```

However, it is then pointed out that the primary purpose of this score data is to determine all the students that do not satisfy a certain threshold score.

As such, you instead propose the creation of n Binary Search Trees. Each node in each tree is to hold:

- `student_ID` in the form "YXXX", where Y is an English letter and X is a digit
- `score` for Subject k (where $1 \leq k \leq n$) and where score ranges from 0.0 to 100.0

Each tree should contain the following methods:

- `initialisation (STRING)`
 - o reads and stores the data from a text file (name given as the STRING input) – i.e., the `student_ids` and scores corresponding to the 1 subject linked to this tree
- `insert (STRING, FLOAT)`
 - o stores the given `student_id` (STRING) and score (FLOAT) in the tree
 - o the initialisation method should utilise this method
- `inorder_traversal (FLOAT): ARRAY<(STRING, FLOAT)>`
 - o performs a special inorder where only scores below the input value (FLOAT) are returned as an array in the form
- `get_all_data(): ARRAY<(STRING, FLOAT)>`
 - o returns an array of all the `student_id` and score data for the subject in question using inorder traversal
 - o this array should be sorted in ascending order of score (i.e., lowest to highest that are under the threshold)
 - o this should correspond to an array of `student_id` (STRING) and score (FLOAT) tuples
- `get_weak_scores (FLOAT): ARRAY<(STRING, FLOAT)>`
 - o prints all the `student_ids` and scores that are under the given threshold (FLOAT)
 - o this array should be sorted in ascending order of score (i.e., lowest to highest that are under the threshold)
 - o this should correspond to an array of `student_id` (STRING) and score (FLOAT) tuples

- 1e Write the code to:

- Generate 4 text files containing the score data of the same 50 students (i.e., the students generated for each of the 4 files should have the same `student_ids`). Use a uniform distribution to generate the individual scores.
- Initialise the required Binary Search Trees (based on your implementation in 1d) to store the data in those text files
- Using thresholds of 40, generate output (to screen – i.e., print) for each Binary Search Tree based on:
 - o `get_all_data(): ARRAY<(STRING, FLOAT)>`
 - o `get_weak_scores (FLOAT): ARRAY<(STRING, FLOAT)>`

- 1f Write code to print out the list of students who did not satisfy their respective subject thresholds, but this time sorted by the number of subjects whose thresholds they failed to satisfy.