FIT2004: Lab questions for week 3

Objectives: This prac allows you to explore the concepts learnt in week 1& 2: abstract data types, operations on ADTs, program verification.
NOTE: This prac is **NOT** assessed. You are required to write all programs in Python.

1. Implement the insertion sort algorithm.
2. Implement the binary search algorithm. Since it is very easy to get the details of binary search wrong when coding it so it is a good idea to test it on at least the following cases:
   (a) an empty array!
   (b) an array of one element,
      i. equal to the key or
      ii. less than it or
      iii. greater than it
   (c) several elements with the key less than the 1st or
   (d) the key greater than the last element
   (e) the key equal to the 1st, a middle, or the last element
   (f) the key not in the array and falling between various positions
3. Let `A[...]` be an array of N floating point numbers; N might be very large indeed. `A[...]` contains some (volatile) floating-point data. We want to write a method to "smooth" this data.
   (i) Write efficient program to print a smoothed version of the data in `A[...]`, smoothing over a sliding "window" of width w, where w is an odd, positive integer. E.g. if $w = 3$, the data under the window [2.0 2.0 5.0] is smoothened to 3.0. Similarly, [2.0 5.0 5.0] $\rightarrow$ 4.0, [5.0 5.0 8.0] $\rightarrow$ 6.0, etc. To deal with the ends of `A[...]`, pretend that it is extended to the left with copies of `A[0]`, and is extended to the right with copies of `A[N-1]`, e.g.

```
For w = 3:
            A[]=  | 2.0 5.0 5.0|8.0 2.0 2.0 5.0  5.0 5.0
   smoothed A[]=  ↓ 3.0(4.0 6.0 5.0 4.0 3.0 4.0
```

   (ii) Give a logical argument why your solution to part (i)
      a) terminates   b) correct.

4. If you still have some time left, and earger to program further, attempt to implement the tree abstract data type (ADT). (Refer to Lecture 1.2). Your program should also implement its basic tree operations: `empty(...)`, `leaf(...)`, `fork(...)`, `left(...)`, `right(...)`, `contents(...)`, `height(...)`, `weight(...)` discussed in the lecture.

```
-=0=-
 END
-=0=-
```