

## Unit 13 Exercises – Sorting and Complexity

### Aims and objectives

- Sorting
- Program planning; coding to specification

### This week's exercises

These exercises are based on ideas covered in recent lectures which you should re-read. You will find it useful to do some planning on paper before you start working on the machines.

### Task 0 – Predicting the outcome of python code snippets (optional)

Go to <http://142.93.32.146/index.php/476898?lang=en> and work through the exercises there.

- these exercises are not part of the course.
- they are entirely optional, yet relevant if you want to improve your programming skills.
- the results are anonymous.
- a group of academics (Fionnuala Johnson, Stephen McQuistin, John O'Donnell and John Williamson) hope to publish anonymized findings based on the collected results.

### Task 1 – Quick sort using recursion and list comprehension

In the lecture, you learned an implementation of QuickSort that uses recursion. Modify the implementation presented in the lecture to use list comprehension alongside recursion. In this implementation, you can pick any element in the list as a pivot.

### Task 2 – Recursive implementation of binary search

In the lecture, you learned an implementation of binary search that uses iterations (i.e., loops). Your task is to implement a recursive binary search.

### Task 3 – First and last occurrence in linear and binary search

a) implement a function that returns the indexes of both: the first occurrence and the last occurrence of an integer in a sorted list of integers. For example:

```
>>> mylist = [1, 3, 6, 7, 7, 7, 88, 103, 426]
>>> input = 7
>>> first_and_last(mylist, input)
First occurrence = 3
Last occurrence = 5
>>> first_and_last(mylist, 6)
First occurrence: 2
Last occurrence: 2
```

b) Modify the binary search algorithm to return the indexes of both: the first occurrence and the last occurrence of an integer in a sorted list of integers as in the example above.

c) What is the time complexity of each implementation?