

# FIT1045/53 Algorithmic Problem Solving – Workshop 8.

## Objectives

The **objectives of this workshop** are:

- To investigate the “divide and conquer” technique.
- To understand how to implement recursion in Python.
- To understand binary search.
- To implement an algorithm for binary search using iteration.

## Task 1

Write a python function `recursive_reverse(a_list)` that takes as input a list and returns a new list which contains the same elements as `a_list` but in the reverse order. You *must* use recursion to solve this problem (i.e. do not use any loops)

## Task 2

1. Write a python function `simple_recursive_power(x, n)` which computes  $x^n$  using recursion. It should require approximately `n` recursive calls to solve  $x^n$ . You *must* use recursion to solve this problem (i.e. do not use any loops)
2. Write a python function `advanced_recursive_power(x, n)` which computes  $x^n$  using recursion. It should require `n/2` recursive calls to solve  $x^n$ . You *must* use recursion to solve this problem (i.e. do not use any loops)

## Task 3

1. Write a python function `simple_recursive_binary_search(a_list, target)` which returns `True` if `target` is in `a_list` and `False` otherwise. You *must* use recursion to solve this problem (i.e. do not use any loops)
2. Write a python function `advanced_recursive_binary_search(a_list, target, lo, hi)` which returns the index of an instance of `target` in `a_list` between `lo` and `hi`, or `None` if `target` is not in that range. When first calling the function, `lo` should be 0 and `hi` should be `len(a_list) - 1`. You *must* use recursion to solve this problem (i.e. do not use any loops)
3. (Optional) How could we change the solution for part 2 to not require `lo` and `hi` to be passed in by the user for the first call?