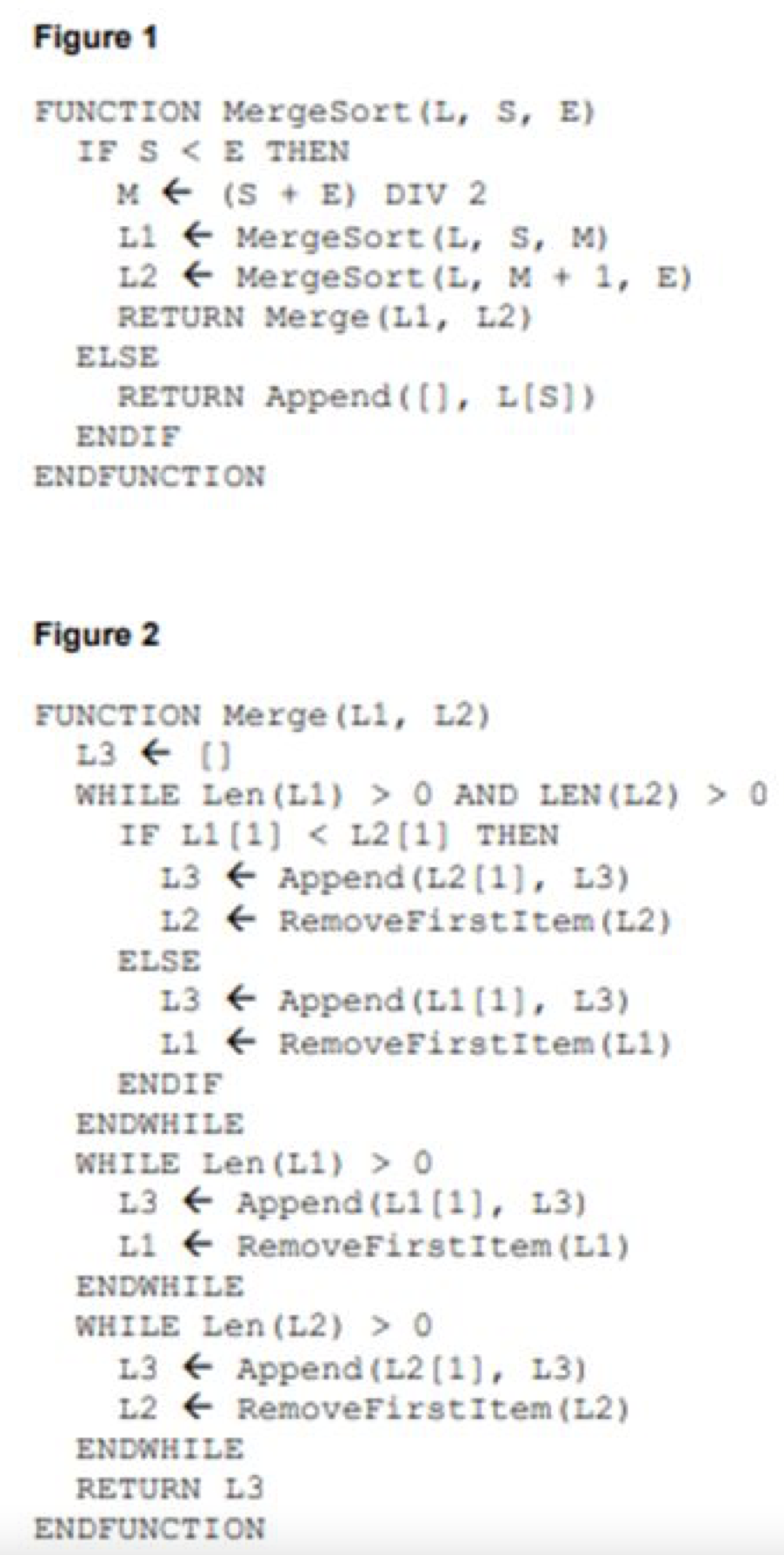
# Big O Notation

The following is code for the Merge Sort Algorithm



**What is the time complexity for Merge Sort?**

**What is the time complexity for Linear Search?**

**What is the time complexity for Binary Search?**

**Time complexity is one measure for describing the complexity of an algorithm. What is the other one?**

**Text, letter

Description automatically generated**

**What does the algorithm above do? What is it’s time complexity and why?**

Table

Description automatically generated

**Which of the algorithms above has O(n log n) complexity?**

**Fibonacci Sequence.**

A Fibonacci number is the sum of the previous 2 numbers where fib(0) = 0 and fib(1) = 1.

One algorithm to calculate fib is shown below:

def fib(n):  
 if n == 0:  
 return 0  
 elif n == 1:  
 return 1  
 else:  
 return fib(n - 1) + fib(n - 2)

**What is the time complexity of this algorithm?**

Copy this algorithm into a Python IDE.

Calculate the following:

fib(0) =

fib(1) =

fib(5) =

fib(10) =

fib(20) =

fib(30) =

fib(40) =

fib(100) =

(Don’t worry about them when they start taking too long)

**How could we improve this algorithm? Work with your neighbour to come up with a way of solving this is O(n) time. No googling allowed**