# Lab 6 – Starting searching and reviewing merge sort

**Part I Starting Searching**

1. Write an algorithm, in pseudo code, which searches for the max. and min. of N numbers in an **unordered** array A. //Bring this to class on Friday

Start Program

Read A

N = A.length

Max = A[0]

Min = A[0]

i = 1;

while i < N

if A[i] > max

max = A[i]

else if A[i] < min

min = A[i]

i = i + 1

Print max

Print min

End Program

1. What is this algorithm’s Big O? Why?

N – as the size of the number in N increases so does the iteration

1. Implement the algorithm in 1. in C.
2. Write an algorithm, in pseudo code, which searches for a particular number in an **ordered** array A. //Bring this to class on Friday
3. What is this algorithm’s Big O? Why?
4. Implement the algorithm in 4. in C.

**Part II Looking at the Merge part of the Merge Sort**

1. Read the presentation from last week on Brightspace. This outlines the **Merge** part of the Merge Sort algorithm. Illustrate/draw how the **Merge** part of the Merge Sort Algorithm works on the following two lists – [1,2,5,7] and [3, 4, 6, 8].