Chapter 2 Algorithms

Antonio Maldonado
September 11, 2016
CS 111

Assume you have a list of numbers

1. Design an algorithm to find the second largest element of the list.

Largest = A[1]

Second Lg = 0

Row = 1

While (row < or = Length ())

{ If (A[row] > Largest)

{ [Second largest = Largest

[Largest = A[row]] }

Row = Row +1

}

Print Second Largest.

2. If there are N elements in the original list, how much time will it take to find the second largest element?

- It would take 0(n^2) time to find the second largest element

3. Assume your list is now sorted from smallest to largest, how long will it take you to find the second largest?

- It would take 0(N log (N) + N) time to find the second largest number.

4. Assume your list is sorted AND you can access any element directly, how long will it take you to find the second largest element?

- It would take N log (N) to find the number.