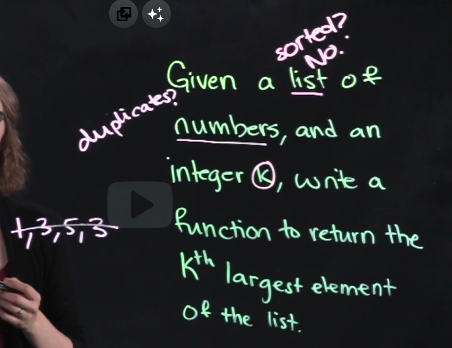
Technical Interview

Part 1: Hearing the problem

Given a list of numbers, and an integer k, write a function to return the kth largest element of the list

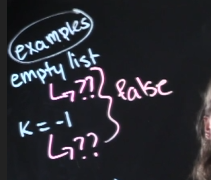
Part 2: Clarifications

1. Is the list sorted?
2. What kind of numbers? Is it integers, naturals, etc.
   1. Are there duplicates in our list?
3. Is k within the bounds of the list? Can k be negative?

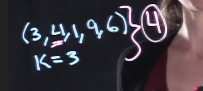


Part 3: Solution & Steps

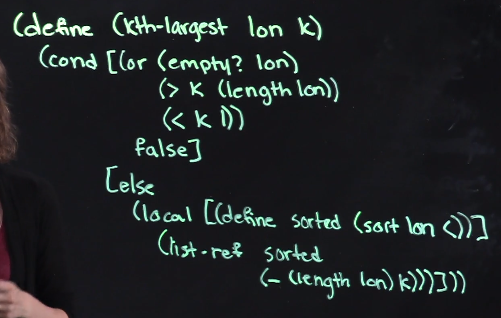
Reflective examples:



* Empty list & K out of bounds
  + Both of these will return false



* Happy path

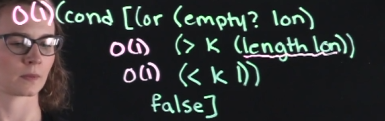


1. Handle base cases first – return false
2. Sort the list
3. Get the largest based on k

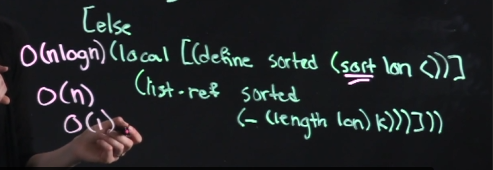
Note: it is totally fine to use functions that you are not sure if it is existing in the current language. They just need to have an overview of the logic of your function

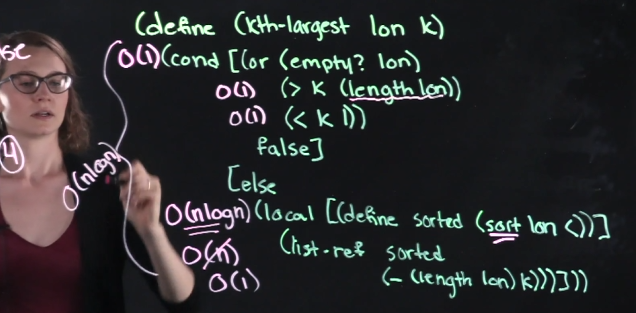
Time complexity

Base case:



Other cases:





Get biggest time complexity

Other ways to solve this problem

* You just find the element k times and make the list smaller each iteration of k
  + Find the largest per iteration of k and remove it in the list
  + On the last iteration, return the largest number
* This will have a time complexity of:



* + Linear search in a list k times
* 
* 
* 
* 