

ADA 2022 Tutorial 2

February 4 2022

1 Finding Majority

You are given an array of n elements, not necessarily distinct. An element is called *majority* if it appears strictly more than $n/2$ times. Your task is to find a majority if it exists. Importantly, the only operation you are allowed to perform to compare two elements is to test if the two elements are identical or different, i.e. there is no notion of an element being less than or greater than another element (and hence *you cannot sort!*). For instance, say the elements are DNA samples, and you can only test if two samples match, there is no notion of ordering, $<$ or $>$ on the samples.

- a. Design a *divide-and-conquer* algorithm to find the majority element or report there exists none. Your algorithm should perform $\mathcal{O}(n \log n)$ tests.
- b. * Do the above in $\mathcal{O}(n)$ tests.

2 Finding n -th smallest element of two sorted arrays

Suppose you are given two *sorted* arrays A and B , each of size n . Design an $\mathcal{O}(\log n)$ algorithm to find the n -th smallest element of the union of A and B

3 Local minimum in an array

Given an array A , we say that an element $A[i]$ is a local minimum if $A[i] \leq A[i+1]$ and $A[i] \leq A[i-1]$ (we only check the inequality for those elements $A[i+1], A[i-1]$ which exist). In other words, a local minimum is an element which is less than or equal to each of its (at most 2) neighbors. Give an algorithm to find *any* local minimum in an array of n elements by making $\mathcal{O}(\log n)$ comparisons.