XJTLU | Computing

Divide-and-conquer

Questions

Problem 1. Write an algorithm of cost $\Theta(\lg n)$ that, given a sorted table T with n different elements, and two further elements x and y with $x \leq y$, return the number of elements in T that fall between x and y inclusively.

Problem 2. A sequence $A = \{a_1, \ldots, a_n\}$ of length $n \geq 3$ is called *unimodal* if there exists an index p with $1 \leq p \leq n$ such that $a_1 < a_2 < \cdots < a_p$ and $a_p > a_{p+1} > \cdots > a_n$. In other words, the sequence increase up to a_p and then decreases. The elements a_p is called its *top*. Write a divide-and-conquer algorithm that, given a unimodal sequence, finds its top in $O(\lg n)$. Argue that the suggested algorithm has time complexity of $O(\lg n)$, and prove its correctness.

References

Atserias, Albert et al. (2022). *Data Structure and Algorithms Problem Set*. Universitat Politécnica de Catalunya.