

CS 5633: Analysis of Algorithms

Homework 2

1. Design a **divide-and-conquer algorithm** which takes as input an unsorted array of n distinct numbers and returns the index of the smallest number in the array. Provide an English explanation of the algorithm followed by **pseudocode**. Set up a **recurrence relation** for the running time of your algorithm, and prove the Θ -bound on the running time via a proof by induction.

2. Recursion Trees and Induction:

For each recurrence, use a recursion tree to find a good guess of the running time of the algorithm (make your guess as tight as possible). Then prove that your guess was correct by induction. In both cases, assume $n \geq 2$.

(a) $T(n) = 25T(n/5) + n^2$

(b) $T(n) = 4T(n/3) + n^4$

3. Suppose you have an idea for an algorithm which will divide a problem of size n into several subproblems of size $n/3$, and the dividing and combining will take $O(\log n)$ time. If you want to ensure $T(n) = o(n^2)$, then what is the maximum number of subproblems of size $n/3$ you can divide your problem of size n ?