Divide & Conquer, Recurrence

Analysis & Design of Algorithms

29 de abril de 2024

Ejercicio 1. Illustrate the operation of MergeSort on the following array

$$A = [3, 41, 52, 26, 38, 57, 9, 49]$$

Ejercicio 2. Consider the following variation of insertionSort. To sort the vector $A[1..n]$, we recursively sort the vector $A[1..n-1]$ and then insert $A[n]$ into the sorted array $A[1..n-1]$. Write the pseudocode for the above algorithm. Write a recurrence relation for the worst-case of this algorithm. Simplify your recurrence assuming all involved constants are 1. Solve the recurrence.

Ejercicio 3. Consider the following search problem. Input: a sorted array $A[1..n]$, and a number v. Output: an index i such that $v = A[i]$ if v is in A and -1 if v is not in A. The binary search algorithm for this problem finds the middle point of A and compares it with v, discarding half of the sequence and repeating this procedure recursively. Write the pseudocode for the above algorithm. Write a recurrence relation for the worst-case of this algorithm. Solve the recurrence.

Ejercicio 4. For each of the following exercises:

- Solve the problems explicitly, ignoring the floor operator.
- Bound both the upper and lower limits using the previous item. Conclude the order Θ of the recurrence. You can use that it is given all these recurrences are increasing.
- Check, using induction, that the Θ notation found in the previous item is correct. You must do this directly from the definition, without using the results of previous items.
- Check using master Theorem if applicable.

Suppose in each case, that $T(1) = 1$ and $T(0) = 0$.

(a) $T(n) = 2T(\lfloor n/2 \rfloor) + n^2$.

(b) $T(n) = 2T(n-1) + 3n - 2$.

(c) $T(n) = 4T(\lfloor n/2 \rfloor) + n$.

(d) $T(n) = 2T(\lfloor n/2 \rfloor) + n^3$.

(e) $T(n) = 7T(\lfloor n/2 \rfloor) + n^2$.