Divided and Conquer

August 23, 2025

Divided and Conquer

- Divide the problem into sub-problems
- Conquer each sub-problem separately
- Solve the original problem by combining solutions for sub-problems

Merge Sort

MergeSort($A = [a_1, \ldots, a_n]$)

- If $n = 1$ Then Return A
- Let $r \leftarrow \lfloor n/2 \rfloor$
- Let $L \leftarrow [a_1, \ldots, a_r]$ and $R \leftarrow [a_{r+1}, \ldots, a_n]$
- $L' \leftarrow$ MergeSort(L)
- $R' \leftarrow$ MergeSort(R)
- Return Merge(L', R')

Merge Arrays

Merge($[a_1, \ldots, a_\ell], [b_1, \ldots, b_r]$)

- Set $C \leftarrow$ empty array of length $\ell + r$
- Set $i \leftarrow 1$ and $j \leftarrow 1$
- For $k = 1$ to $\ell + r$
  If $(a_i \leq b_j)$ and $(i \leq \ell)$ Then $C_k \leftarrow a_i$; $i \leftarrow i + 1$;
  Else $C_k \leftarrow b_j$; $j \leftarrow j + 1$;
- EndFor
- Return C

Analysis of Merge

Lemma

Assume that $A = [a_1, \ldots, a_\ell]$ *and* $B = [b_1, \ldots, b_r]$ *are sorted arrays. Then the output of Merge*(A, B) *is a sorted array of* (A, B).

The running time of Merge is $O(\ell + r)$.

Correctness of MergeSort

Theorem

MergeSort(A) outputs a sorted array with the same elements as A

Running Time of MergeSort

Let $T(n)$ be the worst-case running time of Mergesort on arrays of size n

An observation: $T(n) \leq T(\lfloor n/2 \rfloor) + T(n - \lfloor n/2 \rfloor) + O(n)$

Running Time of MergeSort

We assume that $n = 2^k$ for now. There is a constant $C > 1$ such that:

- $T(1) \leq C$
- $T(n) \leq 2 \cdot T(n/2) + C \cdot n$

Solving the Recurrence

Recurrence for General Parameters

$$f(n) = a \cdot T(n/b) + f(n)$$

Master Theorem

Theorem

Let $a \geq 1, b > 1$, *and let* $T(n)$ *defined by* $T(1) = \Theta(1)$ *and* $T(n) = a \cdot T(n/b) + f(n)$*:*

- *If* $f(n) = \Theta(n^{\log_b a})$, *then* $T(n) = O(n^{\log_b a} \cdot \log n)$;
- *If there is an* $\epsilon > 0$ *with* $f(n) = O(n^{\log_b a - \epsilon})$, *then* $T(n) = O(n^{\log_b a})$.

Examples

- MergeSort: $a, b = 2$ and $f(n) = \Theta(n)$
- $T(n) = 9 \cdot T(n/3) + n \cdot \log n$

Binary Search

Input

- A sorted array $A = [a_1, \ldots, a_n]$
- A number b

Output

Determine whether $b \in A$.

Thanks!