

Mergesort Algorithm

Bakh Khoussainov

Merging two sorted lists.

Input: $A = a_1, \dots, a_n$ And

$B = b_1, \dots, b_m,$

Two sorted lists.

Output: Sorted list C
that merges A, B .

Example: $A: 1, 4, 7, 11$

$B: 2, 3, 6, 7, 10, 12$

Output :

$C = 1, 2, 3, 4, 6, 7, 10, 11, 12$

Algorithm for merging

Initialize a Current pointer to a_0, b_0 ,
 $C = \emptyset$

While A, B both non empty:

Let a_i, b_j be the values of Current pointer.

Append the smallest of a_i, b_j to C .

Advance the Current pointer in the list from which the smaller element was selected.

Append the left-over elements of the longer list to C .

The Mergesort Algorithm

On input list

$$A = a_1, a_2, \dots, a_n$$

Divide A into two pieces of equal size: X AND Y .

Set

$$X \leftarrow \text{Mergesort}(X)$$

(*)

$$Y \leftarrow \text{Mergesort}(Y)$$

Output $\text{Merge}(X, Y)$.

Correctness is proved by induction on n .

When $n=1$, the list is sorted.

Assume the algorithm is correct for all lists of size $\leq n$.

Let A be a list of size $n+1$.

The algorithm builds X and Y ; both of size $\leq n$.

By inductive hypothesis,
after (*), X and Y
are sorted.

Merging sorted lists
produces a sorted list.

—

Thus, we solved the sorting
problem by a Divide and Conquer
method.