

Merge Problem:

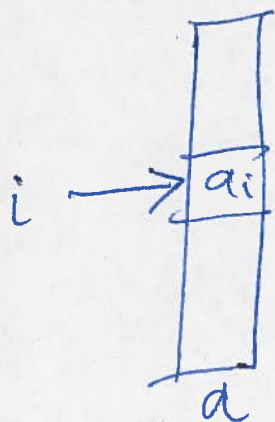
I/p: $\underbrace{a_1, \dots, a_n}_{\text{sorted}} \quad \underbrace{b_1, \dots, b_m}_{\text{sorted}}$

O/p: $2n$ numbers in sorted order.

Eg: $\text{MERGE}(\{1, 3, 5, 7, \dots\}, \{2, 4, 6, 8, \dots\})$
 $= \{1, 2, 3, 4, 5, 6, 7, 8, \dots\}$

Goal: $O(n)$ algorithm
 $O(m+n)$

Init $i \leftarrow 1$
 $j \leftarrow 1$



(*) If $a_i < b_j$
Output a_i
 $i++$

else Output b_j
 $j++$

(*) If $i > n$
Output rest of b 's

(*) else $j > n$
Output rest of a 's.

MergeSort(a, n)

$O(1)$ { If $n=1$ ^{return} ~~output~~ a_1 ,
If $n=2$ ~~output~~ ^{return} $\min(a_1, a_2); \max(a_1, a_2)$

$O(n)$ { $a_L \leftarrow a_1, \dots, a_{\lfloor \frac{n}{2} \rfloor}$
 $a_R \leftarrow a_{\lfloor \frac{n}{2} \rfloor + 1}, \dots, a_n$

return MERGE(MergeSort($a_L, \lfloor \frac{n}{2} \rfloor$),
MergeSort($a_R, n - \lfloor \frac{n}{2} \rfloor$)).

$O(n)$

$T(n) = O(n) + T(\lfloor \frac{n}{2} \rfloor) + T(\lceil \frac{n}{2} \rceil)$

Run time on size n

$$T(n) \leq c \quad \text{if } n \leq 2$$
$$= cn + 2T(\frac{n}{2})$$