Divide & Conquer

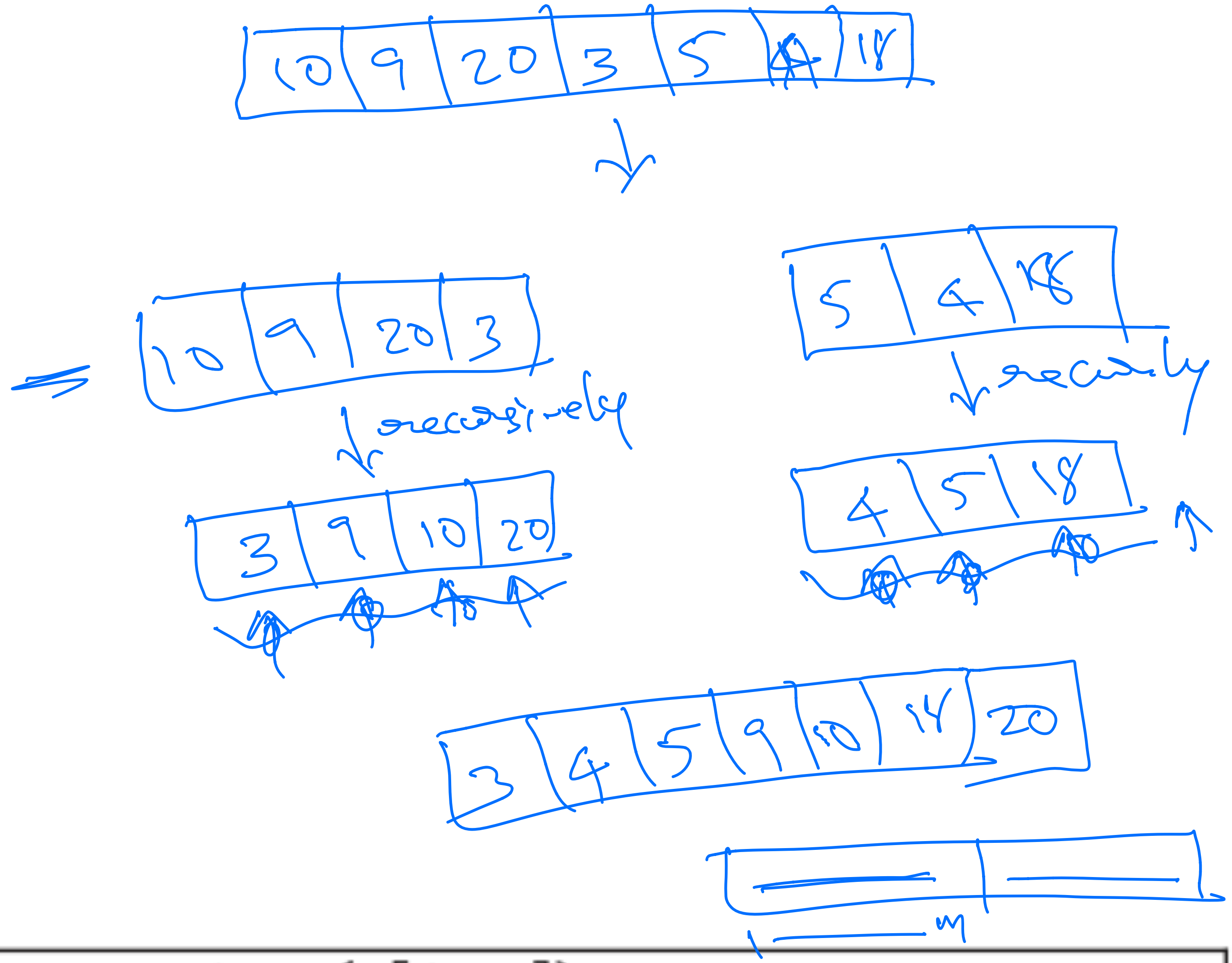
1. Divide into subproblems
2. collect solutions of these sub problems
3. Combine these solutions

- How to solve recurrence relation
    - Recursion tree method
    - Range / Domain transformation
    - Induction / Substitution

Merge Sort

Input: n ~~numbers~~ integers

Output: Sorted array (non-decreasing order)




```
MERGESORT(A[1..n]):
    if n > 1
        m ← ⌊n/2⌋
        MERGESORT(A[1..m])        ⟨⟨Recurse!⟩⟩
        MERGESORT(A[m+1..n])      ⟨⟨Recurse!⟩⟩
        MERGE(A[1..n], m)
```

Lemma: $A[1...m]$ is sorted and $A[m+1....n]$ is sorted then after applying merge$(A[1...n], m)$ $A[1.....n]$ is sorted

```
MERGE(A[1..n], m):
  i ← 1; j ← m+1
  for k ← 1 to n
      if j > n
          B[k] ← A[i]; i ← i+1
      else if i > m
          B[k] ← A[j]; j ← j+1
      else if A[i] < A[j]
          B[k] ← A[i]; i ← i+1
      else
          B[k] ← A[j]; j ← j+1

  for k ← 1 to n
      A[k] ← B[k]
```

A[1 . . . n] - sorted.

Runtime Merge

$O(n)$

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

$$= 2T(n/2) + cn$$

$$= O(n \log n)$$

Exc: Induction
: Recursion
: Range / Domain

Proof

using induction on n

When $n=1$ (base case)

— trivial

◎ $n > 1$

Loop invariant.

⊗ At the end of k^{th} ~~iter~~ iteration:

(a) $B[1 \ldots k]$ are sorted

and the elements of $B[1 \ldots k]$ are

$K[\ldots i-1] \cup A[m+1 \ldots j-1]$

$\{A[1 \ldots$

(b) $A[i]$ and $A[j]$ are greater than or equal to all the elements in $B[1 \ldots k]$

$B[1 \ldots 0] \leftarrow$

Induction $k > 0$.

~~B[1 ...~~

At the end of $(k-1)^{st}$ iteration:

$B[1 \ldots k-1]$ is sorted list of $A[1 \ldots i-1]$

$\cup A[m+1 \ldots j-1]$.

y

if $A[i] < A[j]$

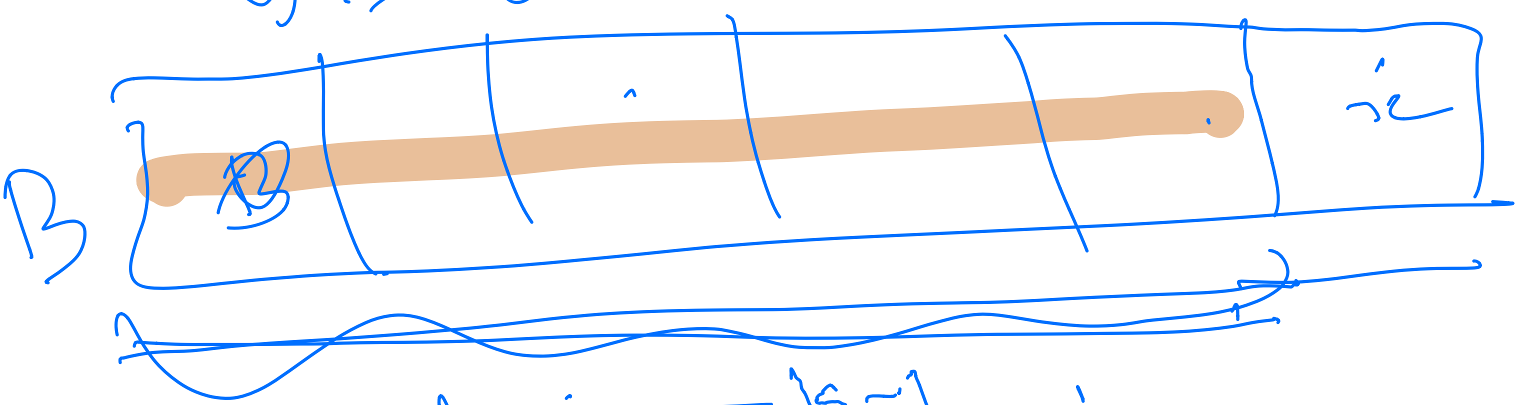
then $A[j]$ is copied

the ith location

$i+1$

$x \geq B[1..k-1]$

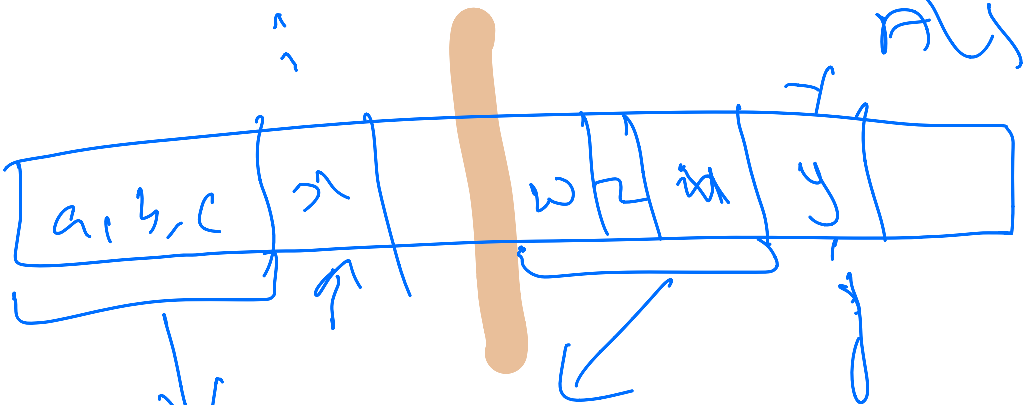
$y \geq B[1..k-1]$

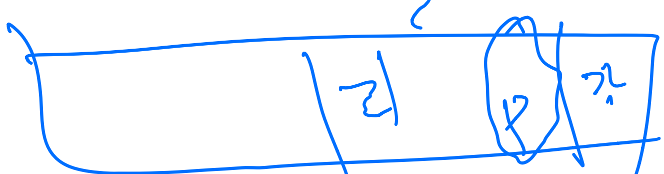


i —— $k-1$

$i = i+1 \longrightarrow A[i] \geq x$

$A[j] = y \geq x$





$x \geq a, b, c$

$x < y$

what is the guarantee that $x \geq z$?

① ~~Right state~~

Come-up with right statements to do a proof by induction.

Quick Sort

Worst-case running time — $O(n^2)$

Average-case — $O(n \log n)$

You assume a probability distribution over the input

$A(n)$ = Average runtime over all inputs of length n.

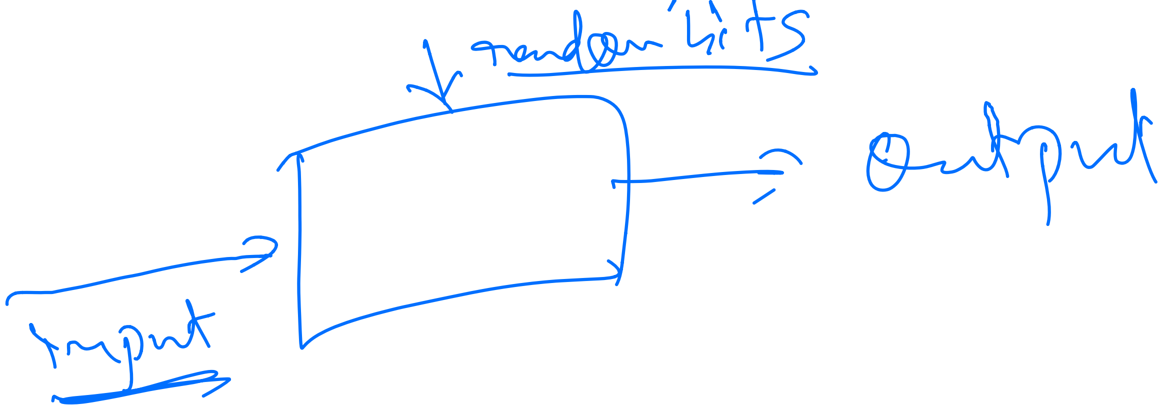
Deter. algos.

"Randomized" Quick Sort.

Worst-case Expected running $= O(n \log n)$

$\uparrow$

$\Rightarrow$ random bits of the algorithm.



$\Rightarrow O(n \log n)$

— $O(n)$ - algorithm for finding median.