You've 2 represent only 2 states

ON | OFF

0 / 1

1000
101
0000

0/1

→ [1 Bit]

bool on[0]; on[0] → _ _ _ _ _ _ _ _

8 bit

Q. Given an array, count the no. of pairs (a_i, a_j) where i, j are indexes and

$i < j$

$a_i > a_j$ ← pair satisfy this

inversion count

9, 2, 1 → 3 ans

mergesort

$O(n \log n)$

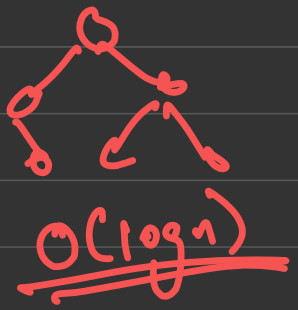
↳ Policy Based Data Structure

↳ Set → O(log n) Red Black Tree

insert delete search

BBST

↳ Trie →



O(log n)

Rb_tree_tag ⟶ find_by_order

⟶ order_of_key

(1) find_by_order(int k) → return iterator to the k^{th} largest no. (counting from 0)

Smallest is 0^{th} largest.

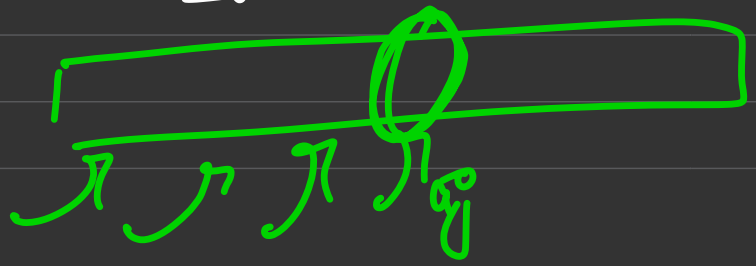
(2) order_of_key(int x) → returns no. of items in a set that are strictly smaller than x.

all the possible valued a_i will lie on the right part of a_j

How many elements till now are greater than a_j

y = # of elements $< a_j$

Simple



$a_i + a_j > b_i + b_j$

$a_i - b_i > -(a_j - b_j)$ for any 2 topics i, j

$i < j$

$c_i > -c_j$

$i < j$

$a_k - b_k = c_k$ modified in count

$c_i = -c_j$ pair

$\{c_k\}$ $\{c_k\}$

PBDS