PS_6

(P1)

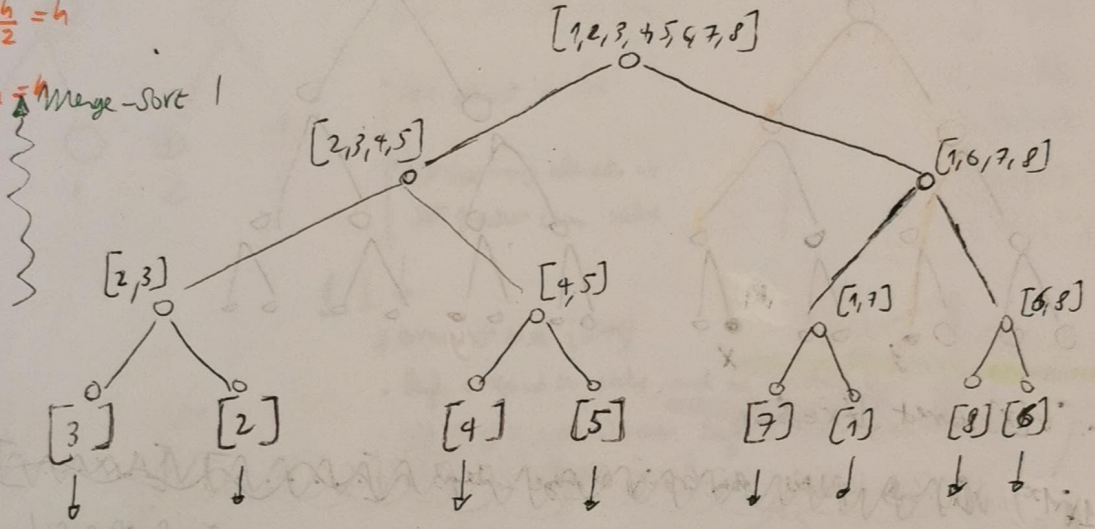
$P_1, P_2, \ldots, P_n$ $\quad 1 \ldots n$ ⟶ inversion vector V in $O(n \log n)$

What need to prove here? (no proof needed)

Algorithm:

$\frac{n}{2} + \frac{n}{2} = n$

$\frac{n}{4} + \frac{n}{4} + \frac{n}{4} + \frac{n}{4} = n$ ← Merge-Sort !



* Vi's: (init. to 0)

merge

	1	2	3	4	5	1	2	3	4
	1	5	7	9		6	8	11	20
i=	0	1	2	3					

l = 4

"we count # swaps to be made"

	1	5	6	7	8	9	11	20
	i=0	i=1	i=2	i=2	i=3			

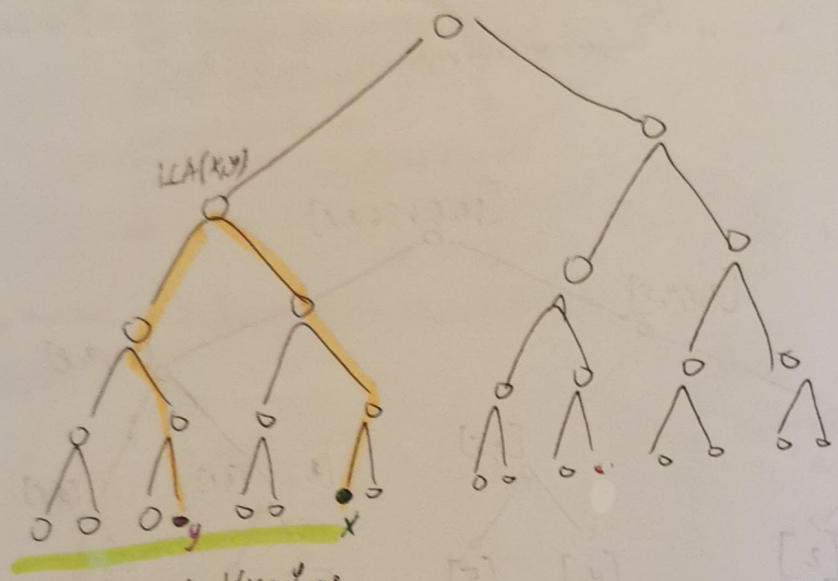
$V[8] += 4 - 3 = 1$

$V[6] += 4 - 2 = 2$

* time complexity

$O(n \log n)$

* Edge: n not a power of 2:



Claim: "v[x] correct $\forall x \in$ leaves"

* When update v[x]? [...] [x]

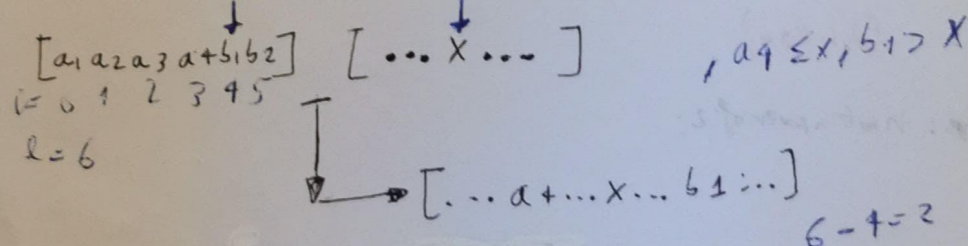
two invariants:

1) "consider an element y" for x := we're merging two intervals [... y ...] [x ...]

1) We only consider elements before x in the original list
(clear from tree structure)

2) $\forall y$ before x, y is considered exactly once:
at the merge which makes LCA(x,y) merged

1 * Whenever we merge with x on the right:



We update v[x] with # elements > x in the left list.