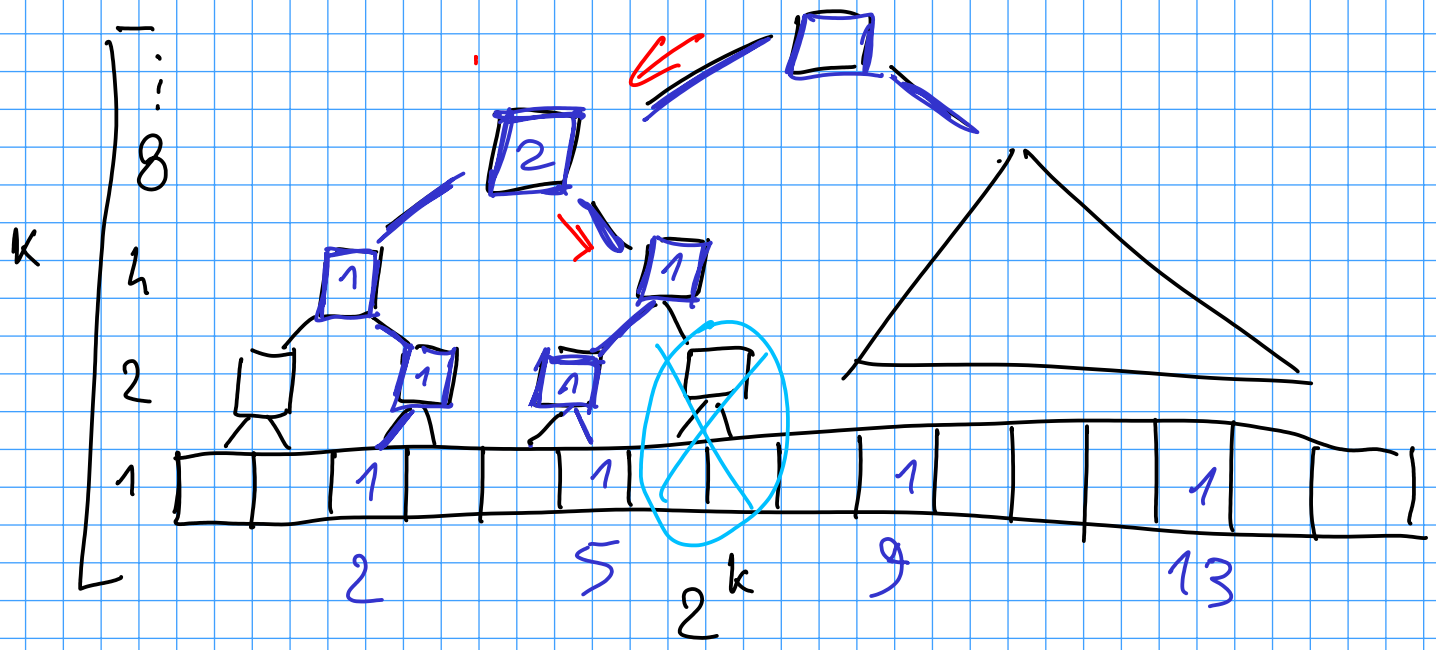



```
struct Node {
    Node * left;  <-   2 i + 1
    Node * right; <-   2 i + 2
    int val;
};
```

SEGMENT SPARSO SU BITMASK

→ molte operazioni ok in set (treap)

complessità : $O(\log K)$ K = range

find, next, find of by rank

Nodo $\Rightarrow$ X, P

X: valore da memorizzare

P: priorità

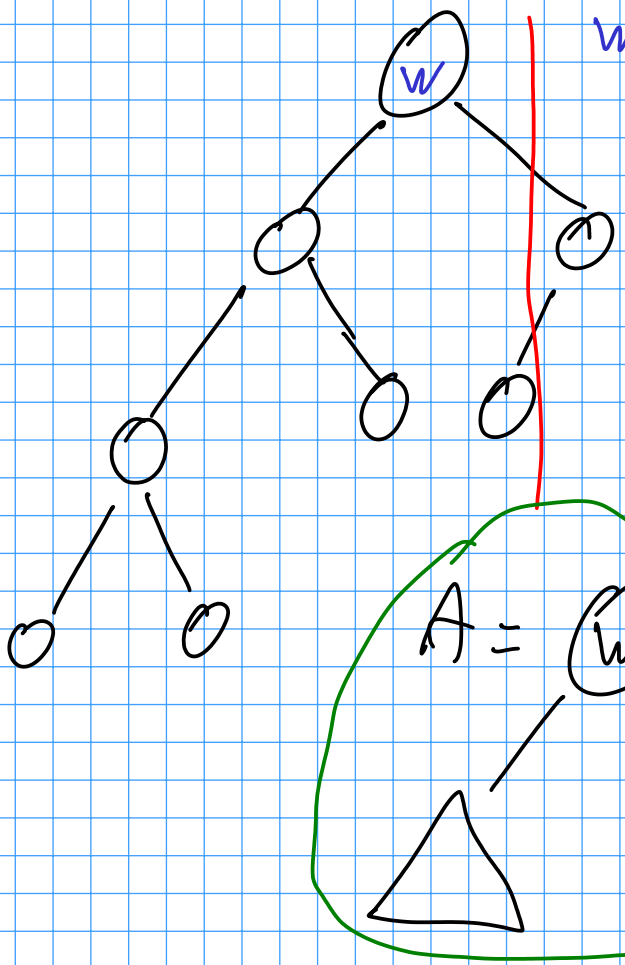
l'albero è un BST su X
e un heap su P

Fatto: se i valori P sono random e distinti, allora il treap è unico.

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

split $(T, X) \Rightarrow (A, B)$ con $a \le x \ \forall a \in A$, $b > x \ \forall b \in B$

merge (A, B) con $x < y \ \forall x \in A, y \in B$

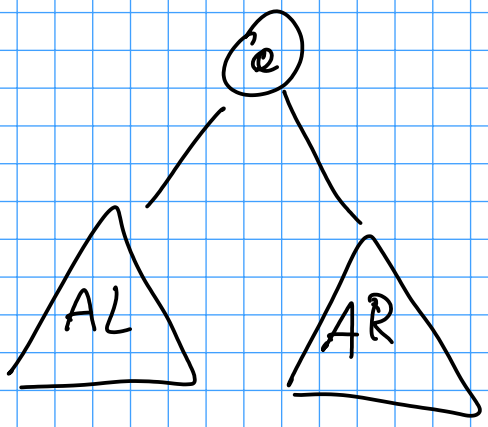


$w \le x \Rightarrow w$ in A
w.left in A

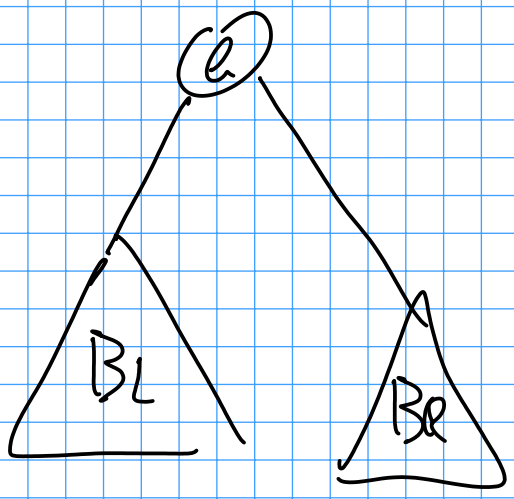
split su w.right
$\Rightarrow A', B'$

$A' \le x \quad B' > x$
$A' > w$

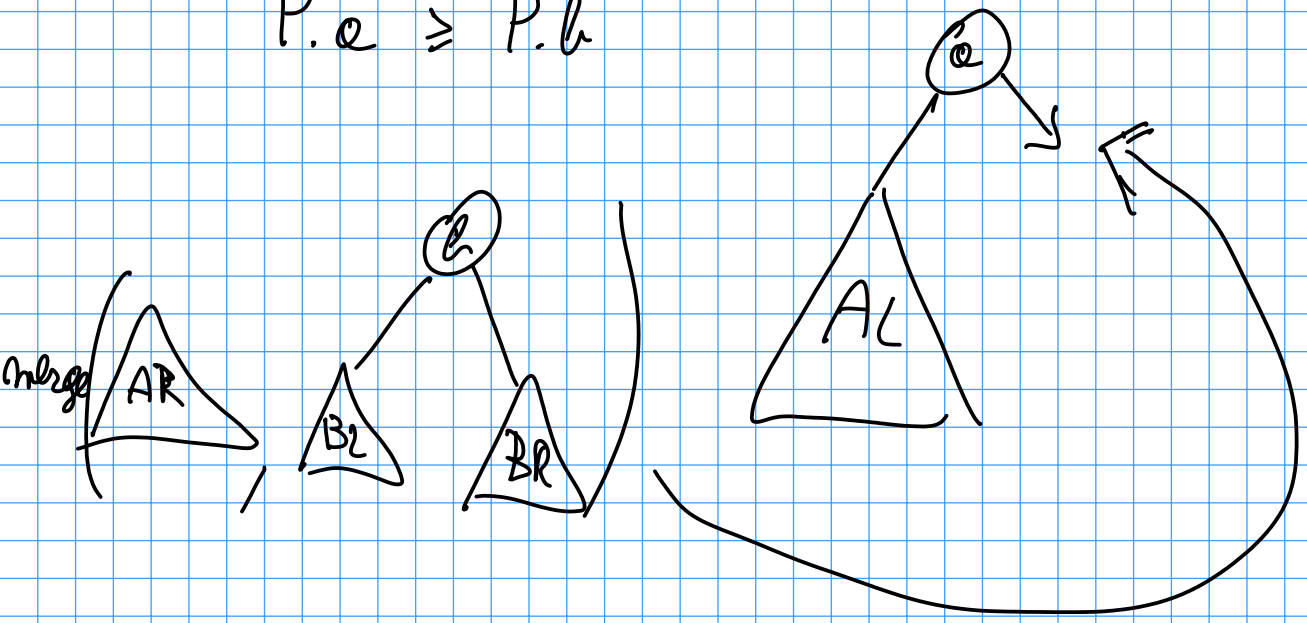
B'



$\leq$



$P.a \geq P.b$



insert (x)

split (x) → A, B

merge (A, merge (x, B))

erase (x)

split (x) → A, B

A. split (x-1) → A', x

merge (A', B) → T

split persistente su treap: