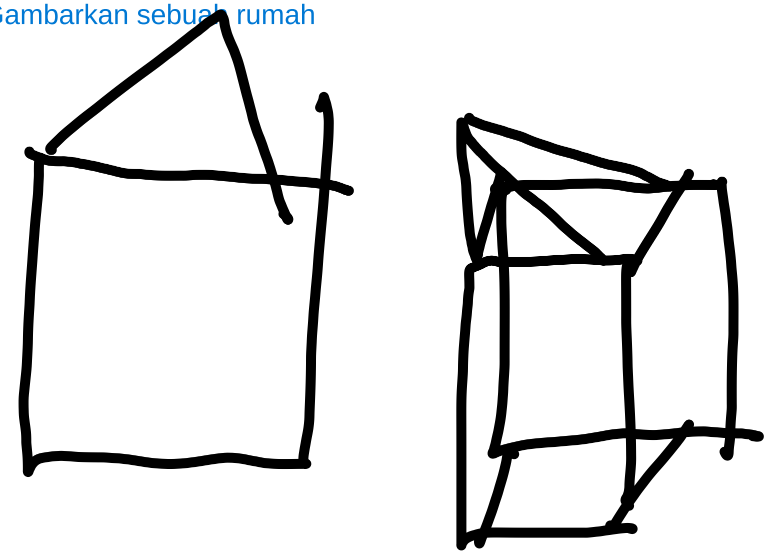
Binary Search Tree

Data Structure RKA

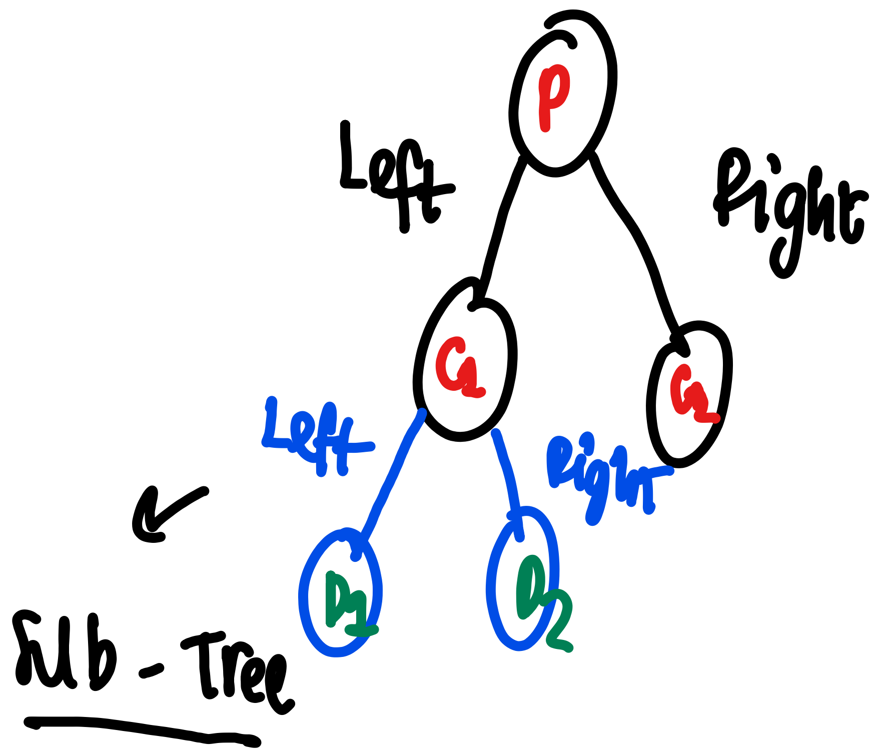
Abstract Data Type $\rightarrow$ Ekstrak'

Gambarkan sebuah rumah



$$\begin{tikzcd}
\text{Tree} \arrow[r] & \text{Parent} \arrow[d] & \\
& \text{Children} \arrow[dl] \arrow[dr] & \\
\text{Left} & & \text{Right}
\end{tikzcd}$$

$$\begin{tikzcd}
\text{Tree} \arrow[d] \\
\text{Sub Tree}
\end{tikzcd}$$



```
P = {
  left: C1 :{
         left:D1,
         right:D2
   }
  right: C2
}
```

$$\begin{array}{lccccc} \text{Idx} & L \leftarrow \boxed{0} & 1 & 2 & \boxed{3} \rightarrow R \\ \text{arr} = & [\,\underline{2}, & \underline{4}, & \underline{7}, & \boxed{\underline{9}}\,] \end{array}$$

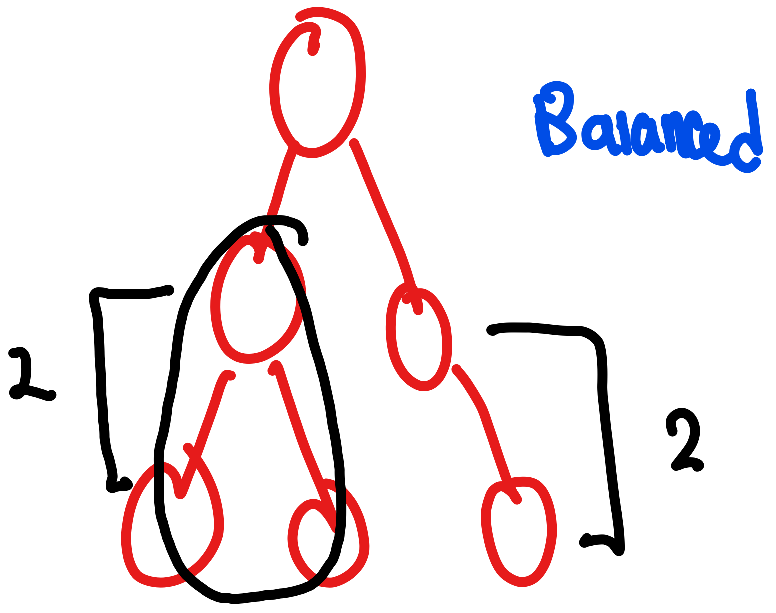
ukuran = $\boxed{4}$ jml array → len(arr)

↓

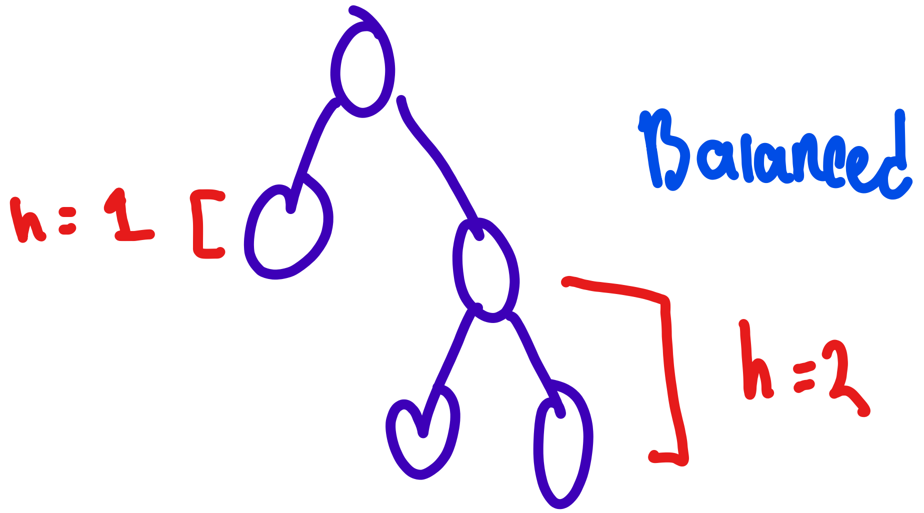
$4-1=3$

$L=0$, $R=\underline{\text{len(arr)} - 1}$

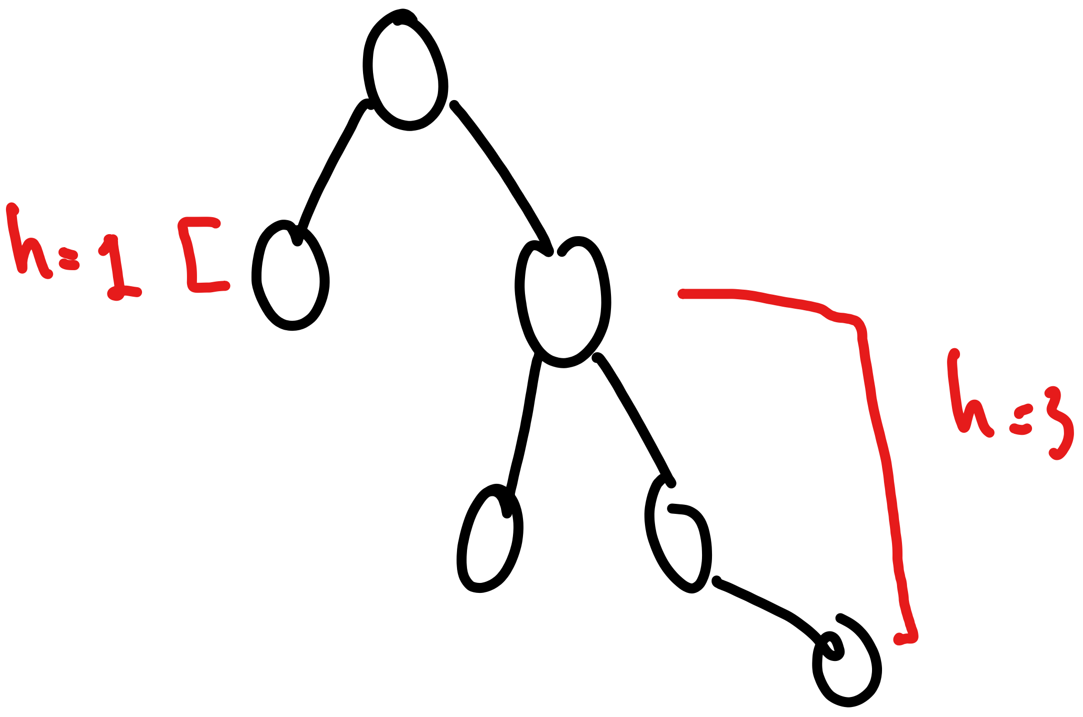
Balanced Binary Tree



$2 - 2 \leq 1$ (V)



$2 - 1 \leq 1$ (V)

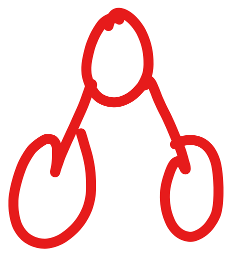


3−1 7 1 (x)

Not balanced

sorted array → len ganjil
len genap

serial sub tree



un balanced →

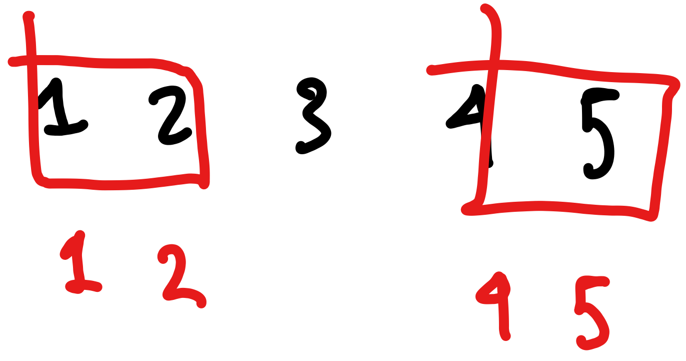
L = 3, ...

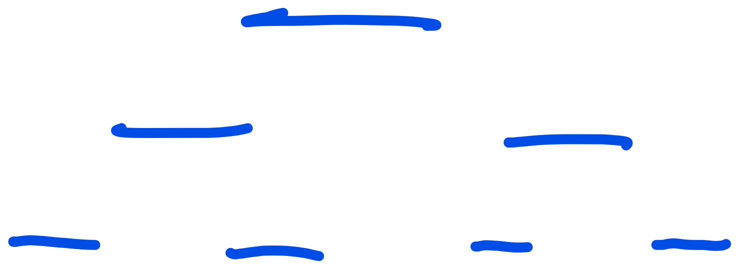
R = 3, ...

$L \le R$

$R - L \le 1$

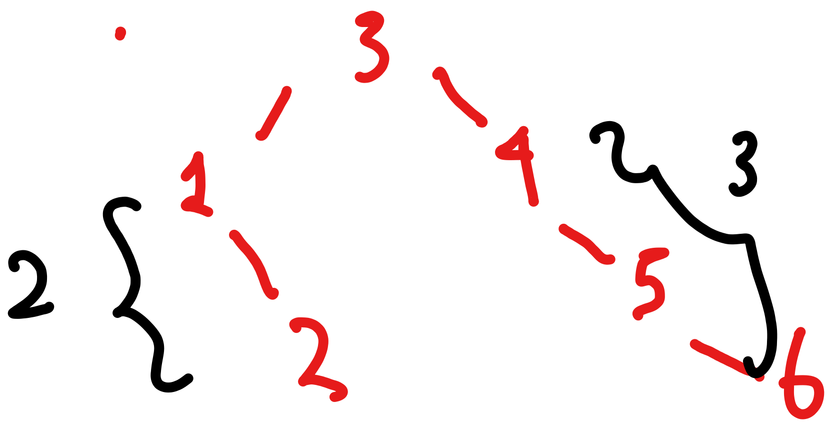
optimal → 1/2 1/2 1/2 → balanced

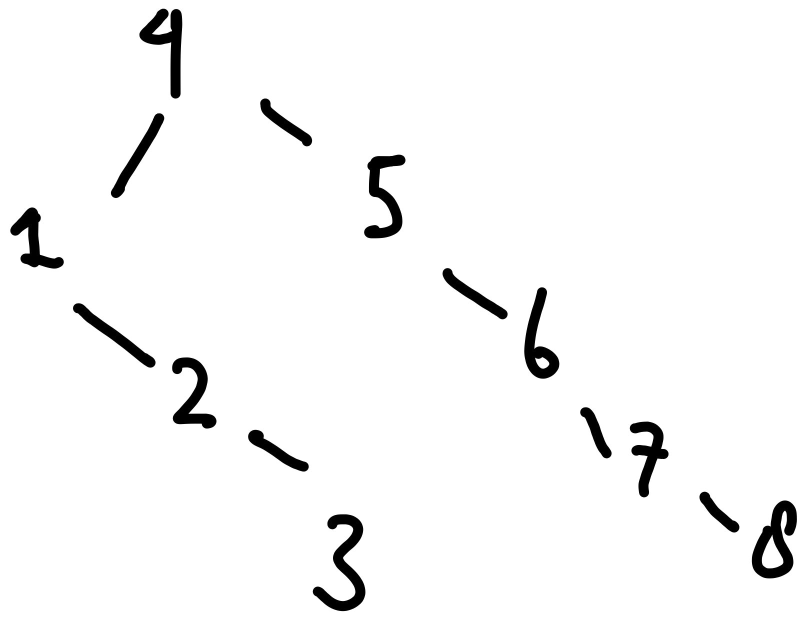




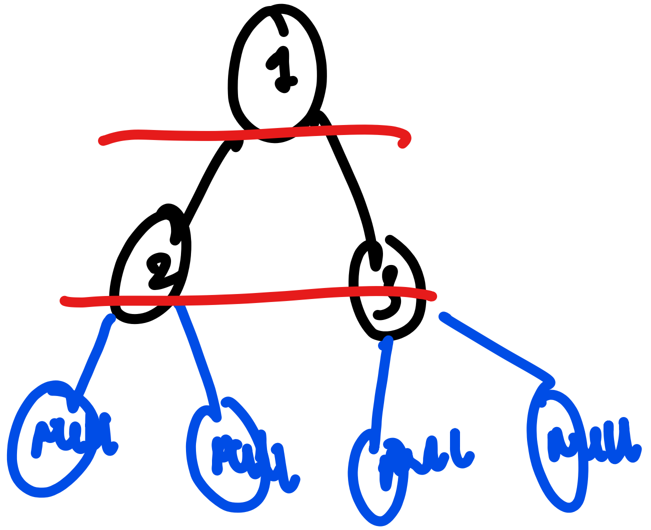
$1\ 2\ \boxed{3}\ \textcircled{4}\ 5\ 6\ 7\ 8$

$3-2 \leq 1$





$1, \textcircled{1}$



Q = [1]

Q = [2,3]

Q = [~~Null~~, Null,

pop

Pop

node = "Null"

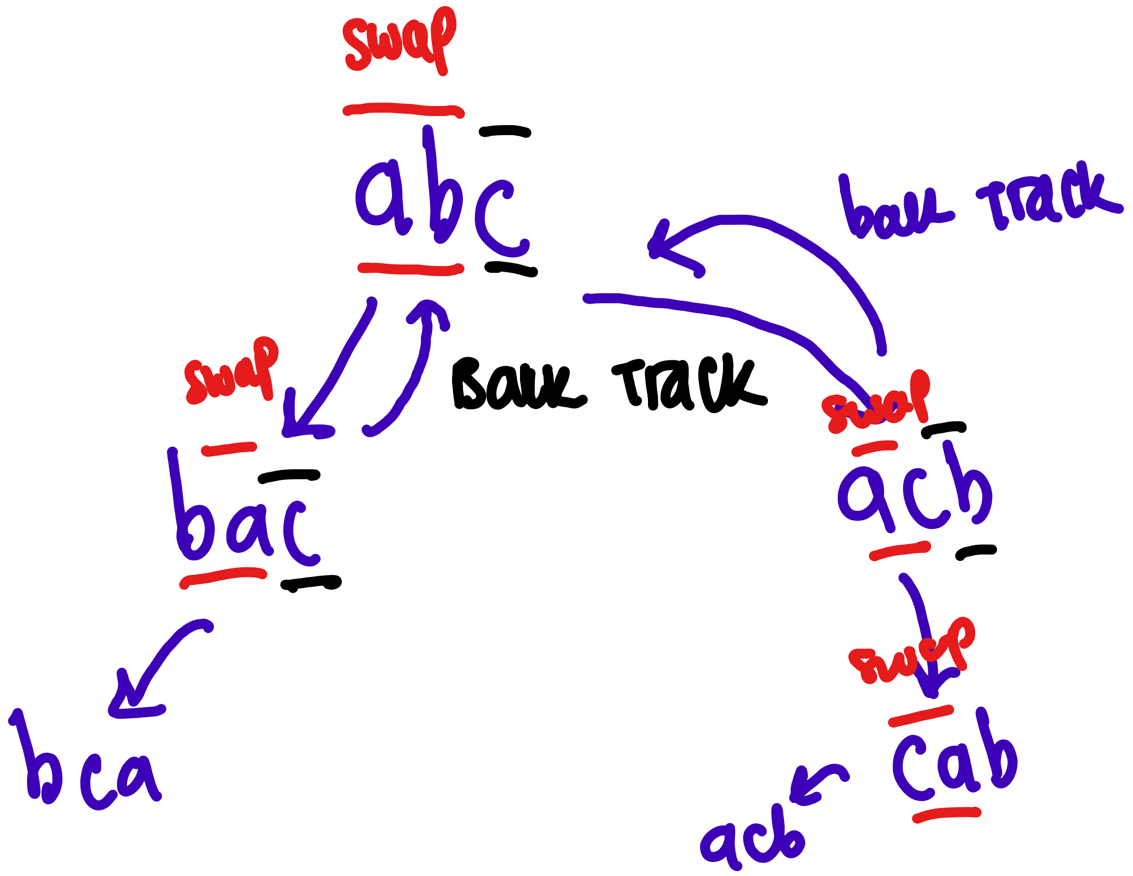
next_node = "Null" ?

Q[0]

Back-Tracking

$$\text{"abc"} \xrightarrow{\text{Permute}}$$

abc, acb, bac, bca
cab, cba



a	b	c
0	1	2

swap(0 , 1)

swap(1, 2)

$$\begin{tikzcd}0 \arrow[d] & 1 \arrow[d] \\ 1 & 2\end{tikzcd}$$

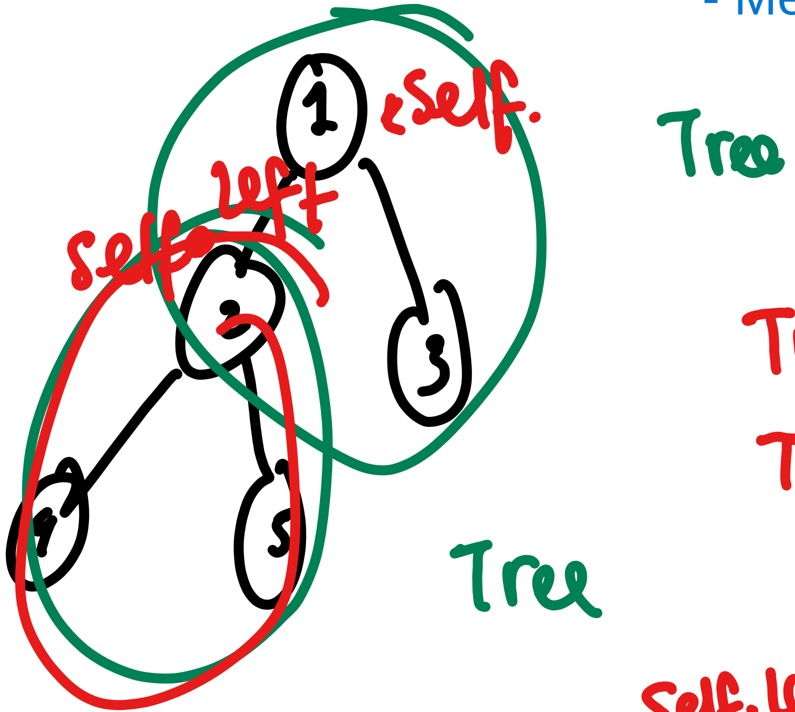
OOP → object

Manusia :
{
  Mata :
  Hidung :
  Mulut :
}

Tree :{
  Left :
  Right:
}

Abstraction

- Atribut
- Method : insert, traverse, delete, searc



Tree1 = Tree(1)

Tree1.Left.insert(2)

Self.left.Val = 2