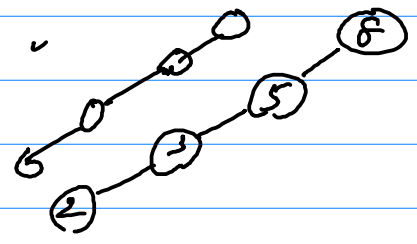
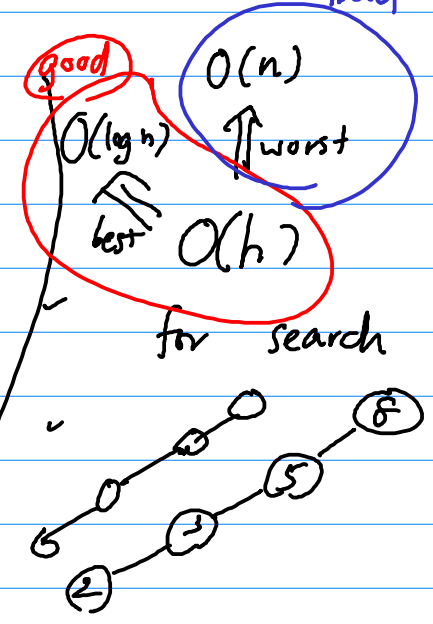


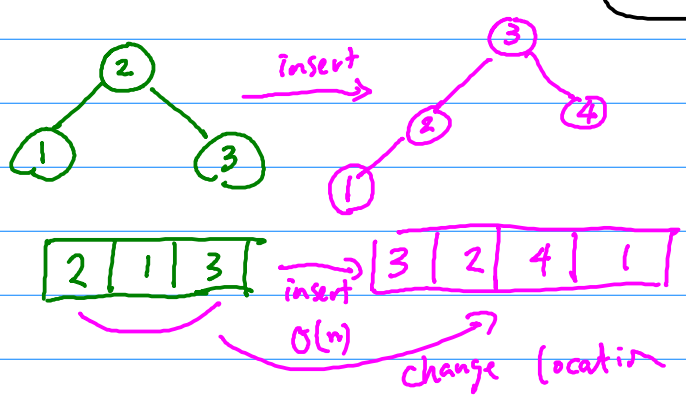
```

BST-Search (k, T) {
  mid ← root of T
  if (k < mid)
    return BST-Search (k, T → left)
  if (k > mid)
    "
    T → right
  else if (" = ")
    return mid.value
}
  
```



⚡

restriction	loose	faster in general	easy to teach	strict
worst search time	$O(n)$	$O(\log n)$	$O(\log n)$	$O(\log n)$
maintenance after insertion	$O(1)$ arbitrary BST	$O(1)$ Red-Black Tree	$O(1)$ AVL Tree	$O(n)$ complete BST
			ordered array	



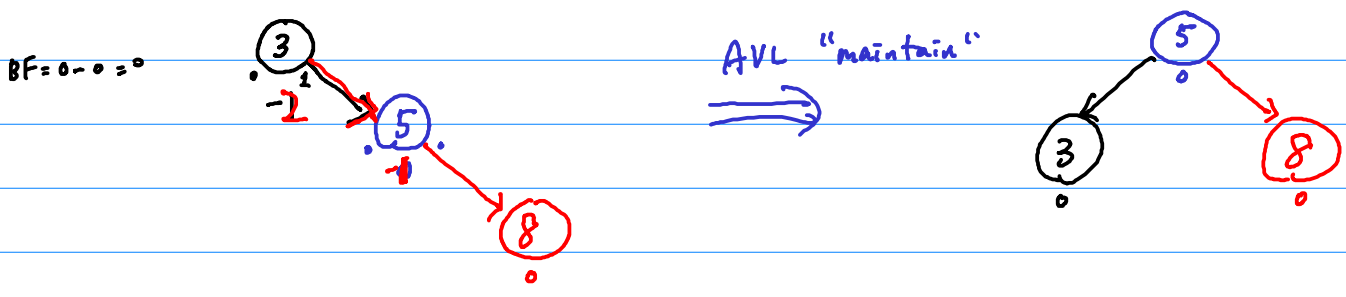
* Adelson Velskii Landis (1962)
a BST such that

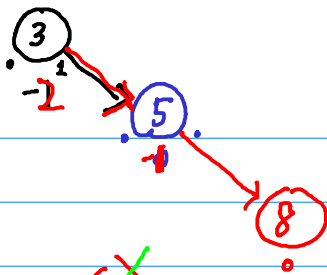
$$\left| \frac{\text{height}(T_L)}{h_L} - \frac{\text{height}(T_R)}{h_R} \right| \leq 1 \quad \text{for every subtree}$$

$$\text{BF}(T) = h_L - h_R \begin{cases} 0 \\ -1 \end{cases}$$

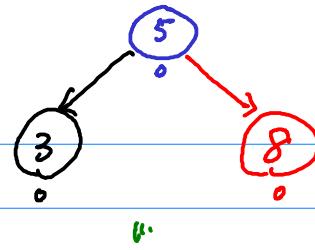
balance factor

* (3) (5) (8)





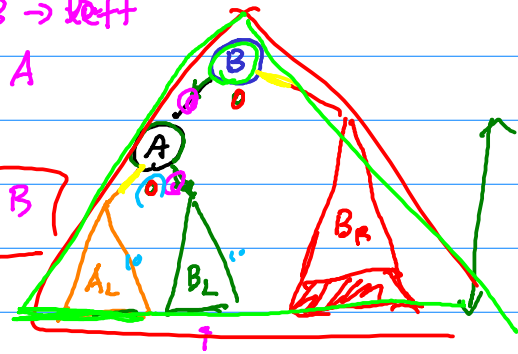
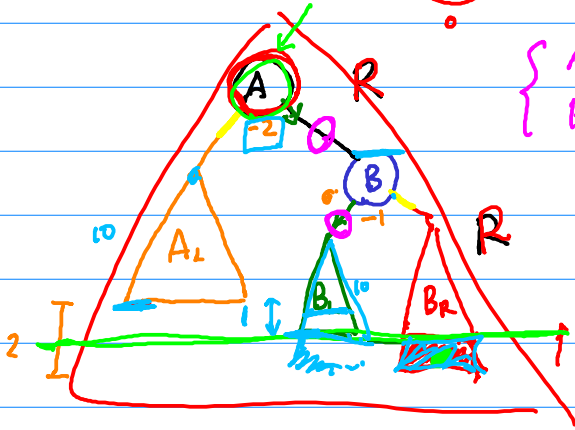
AVL "maintain"



AVL: RR
LL

$\left\{ \begin{array}{l} A \rightarrow \text{right} \leftarrow B \rightarrow \text{left} \\ B \rightarrow \text{left} \leftarrow A \end{array} \right.$

rotate A, B



* 1, 2, 3, 4, 5, 6, 7

