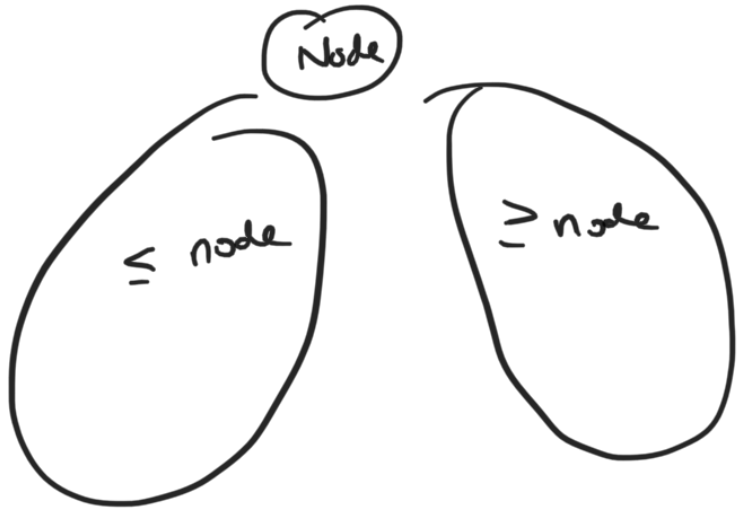
# Trees

Binary search tree (BST)




∀ node

Balanced tree ⇝ ensure O(log n) for insert, find

(eg) Red-Black trees,
AVL trees

full binary tree – every node has 0/2 children

## Binary tree traversal

### in-order

print left branch
print current node
print right branch

BST —in order→ ascending

### pre-order

print current node
print left branch
print right branch

node = first

### post-order

print left branch
print right branch
print current

order

node visited

node

root = last node visited

## Bidirectional search

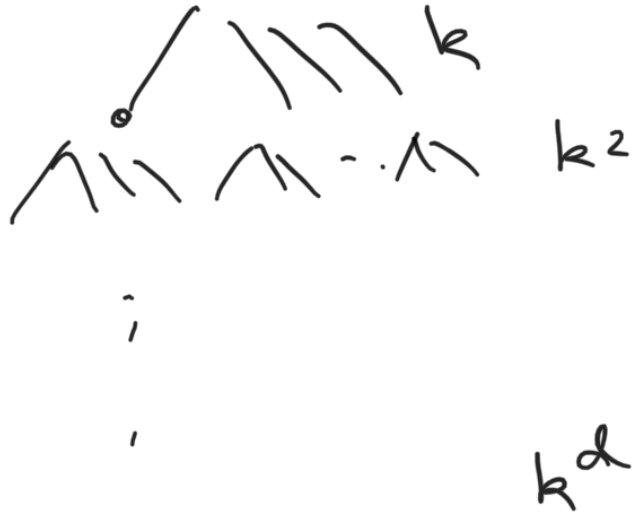
shortest path from $s \to t$

→ Run 2 simultaneous bfs, one from each node $s, t$

→ searches collide,


If $deg(v) \le k$ (traditional bfs)

$s \xrightarrow{d} t$ )




$k + k^2 + \dots + k^d$

$1 + k + \dots + k^d = \frac{k^{d+1} - 1}{k-1} \sim O(k^{d+1})$

in bidirectional

$$2k^{d/2+1} \rightsquigarrow O(k^{d/2+1})$$