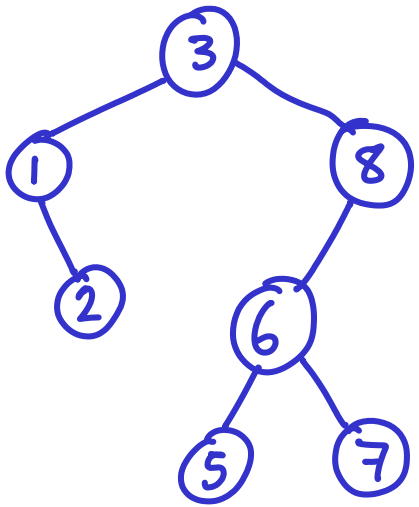
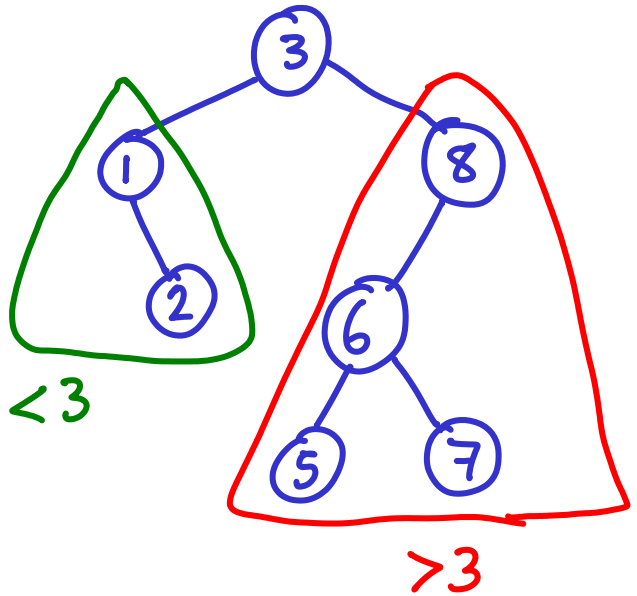
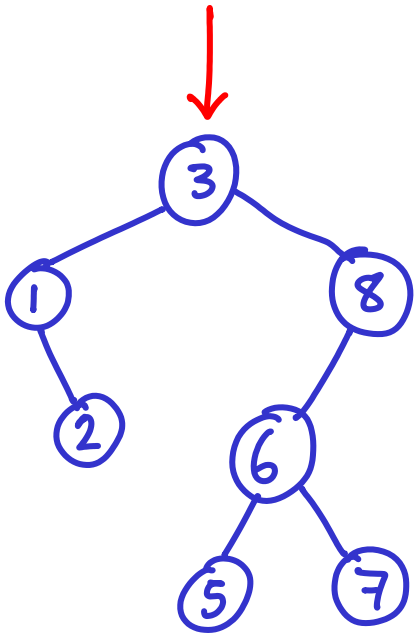
Binary search trees





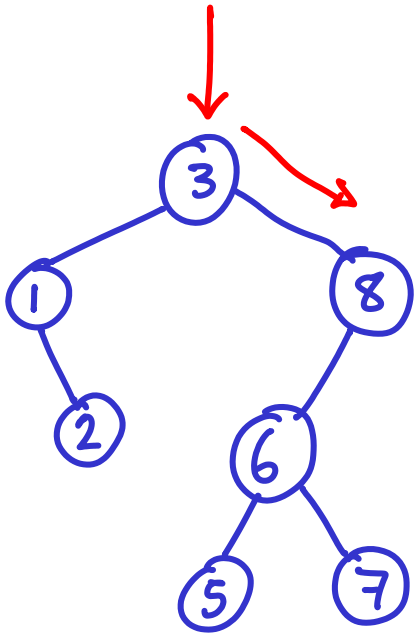
Binary search trees

(binary) search for 5



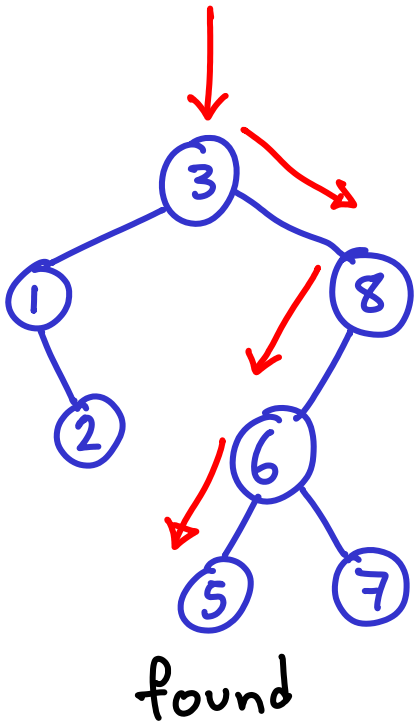
Binary search trees

(binary) search for 5



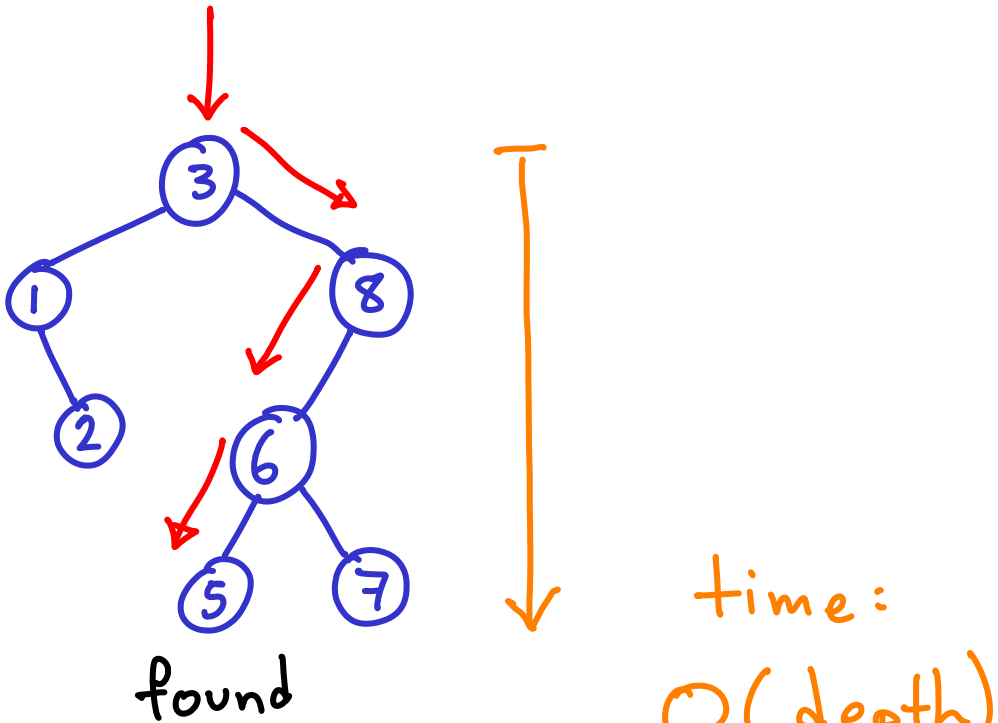
Binary search trees

(binary) search for 5



Binary search trees

(binary) search for 5

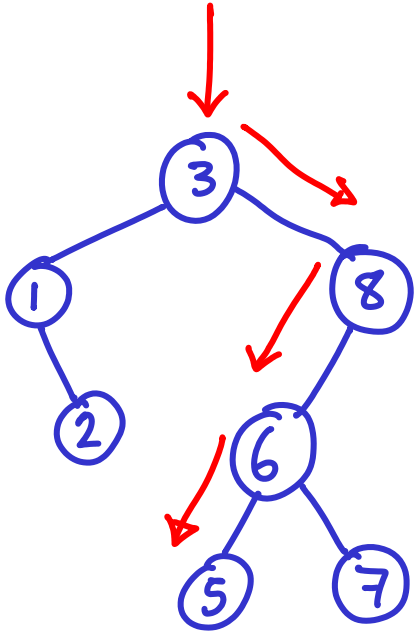


time:

O(depth)

Binary search trees

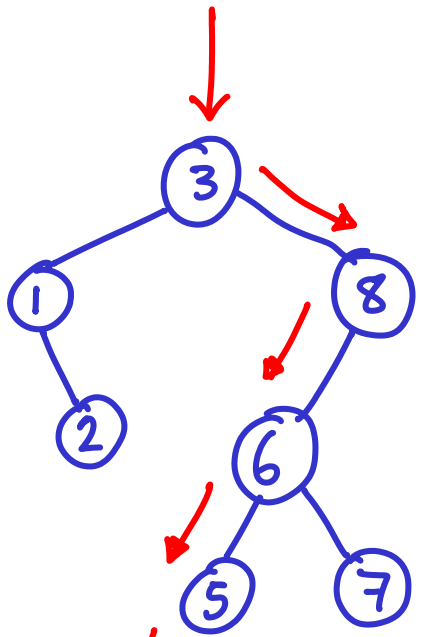
(binary) search for 5



found

time:
O(depth)

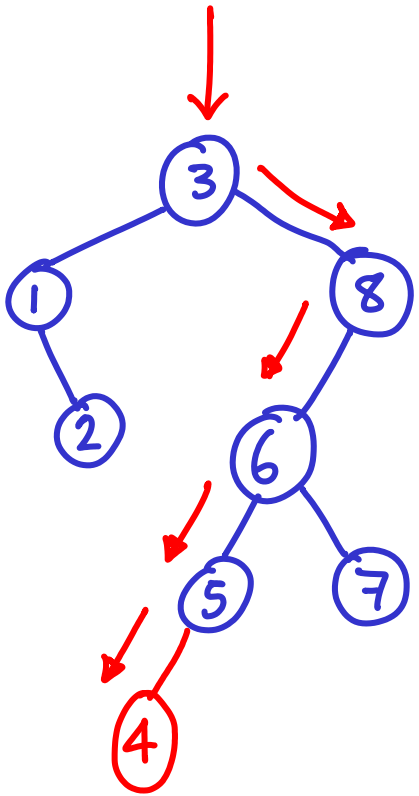
(binary) search for 4



x not found

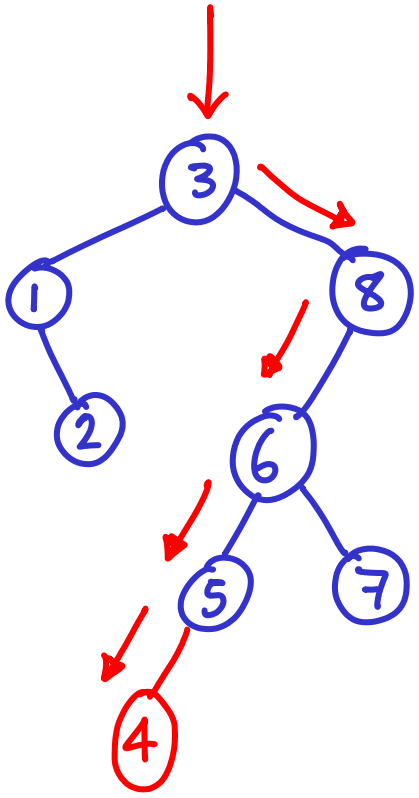
Binary search trees

insert (4) ~ search

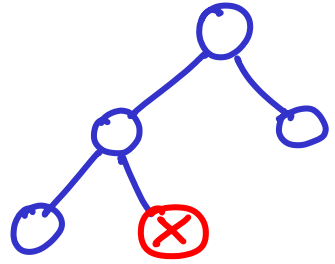


BINARY SEARCH TREES

insert (4) ~ search



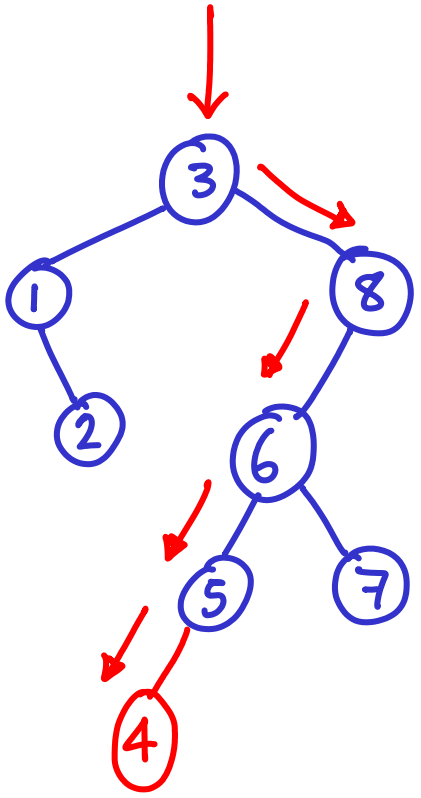
instant delete (x)



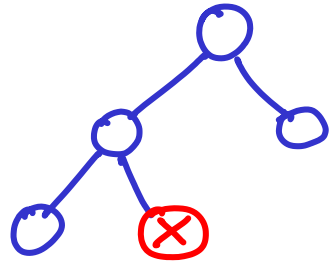
x : leaf

BINARY SEARCH TREES

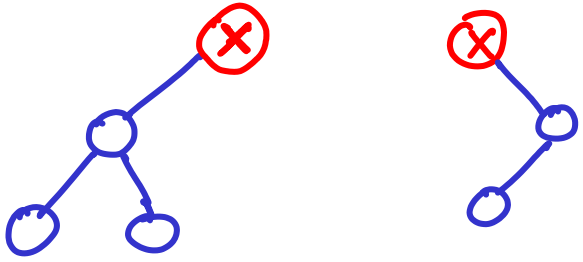
insert (4) ~ search



instant delete (x)



x : leaf

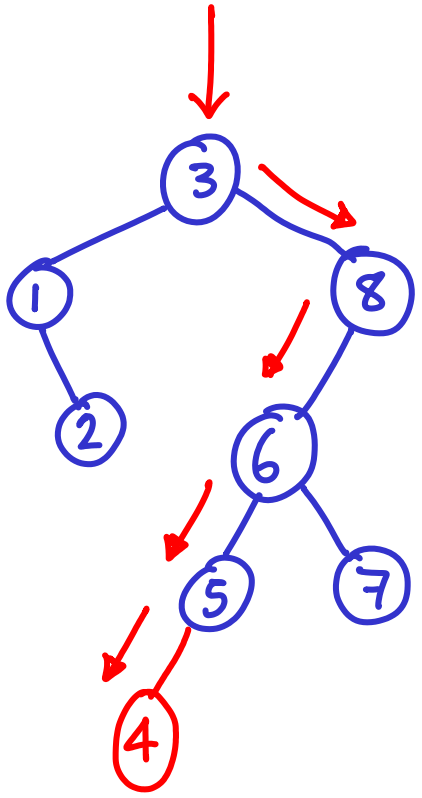


x : root w/ 1 child

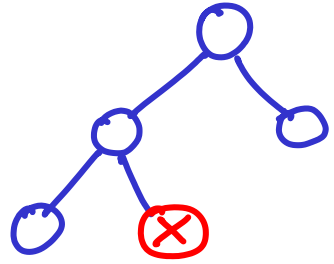
↳ x = MIN OR MAX

BINARY SEARCH TREES

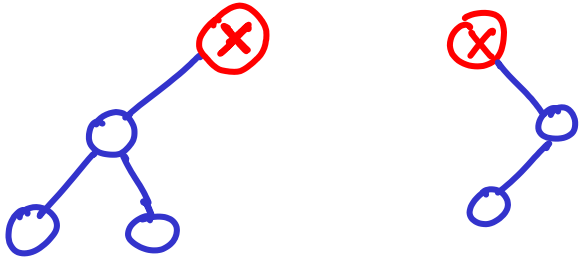
insert (4) ~ search



instant delete (x)

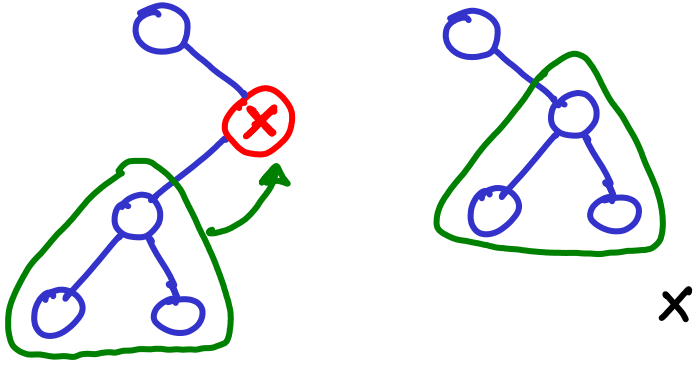


x: leaf



x: root w/ 1 child

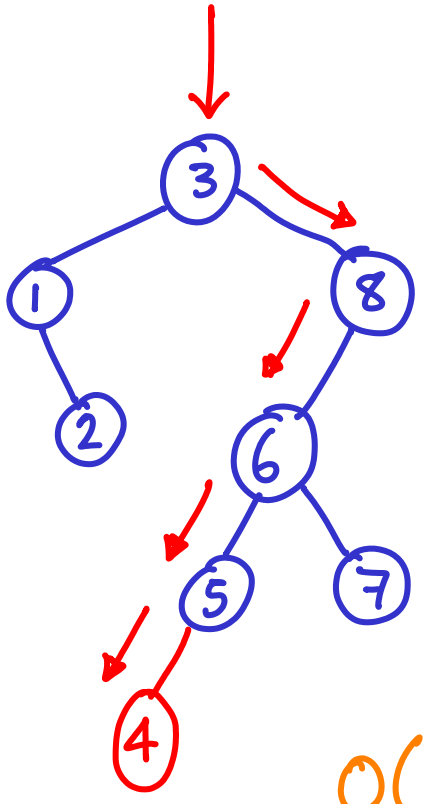
↳ x = MIN OR MAX



x: any node w/ 1 child

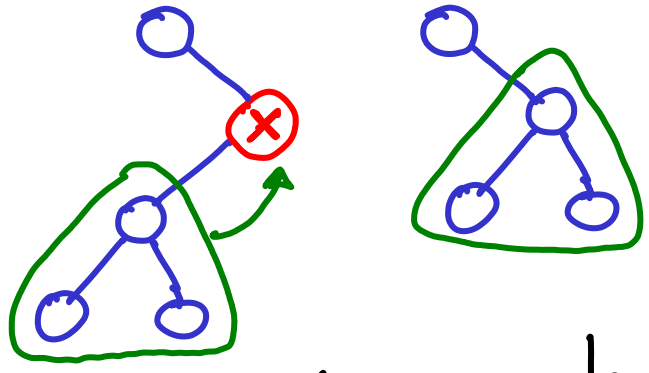
Binary search trees

insert (4) ~ search



O(depth)

instant delete (x)



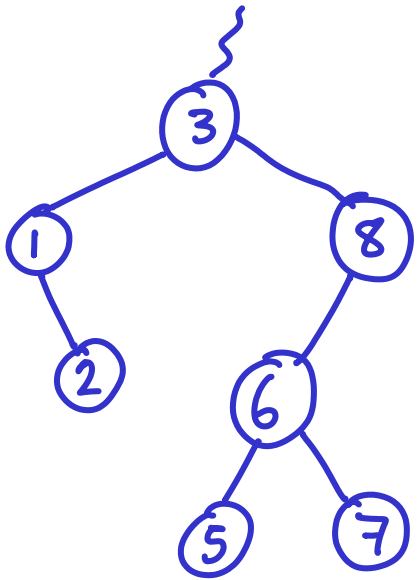
x: any node w/ < 2 children

O(1)

If (one) subtree exists, promote it.

Binary search trees

delete(3)

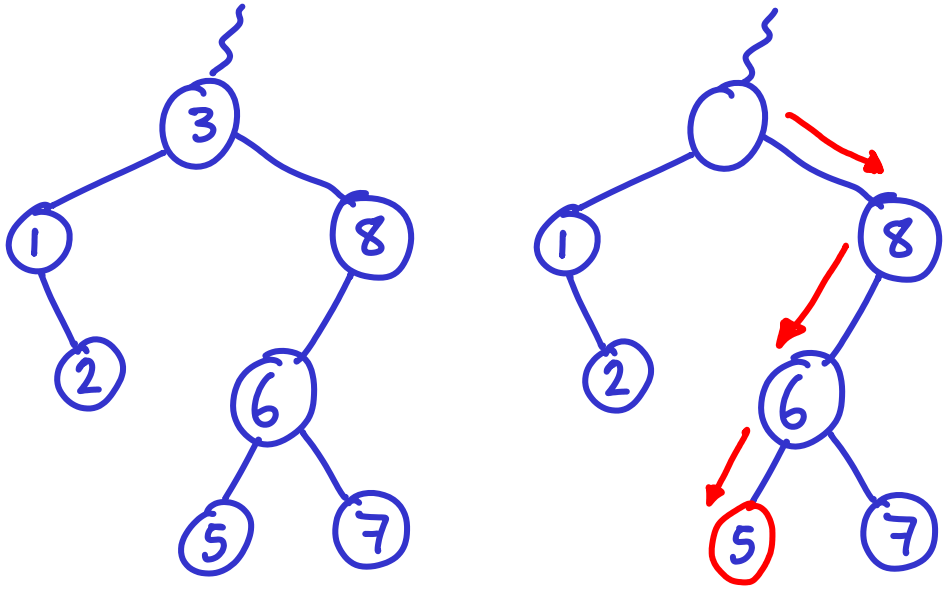


BINARY SEARCH TREES

non-instant

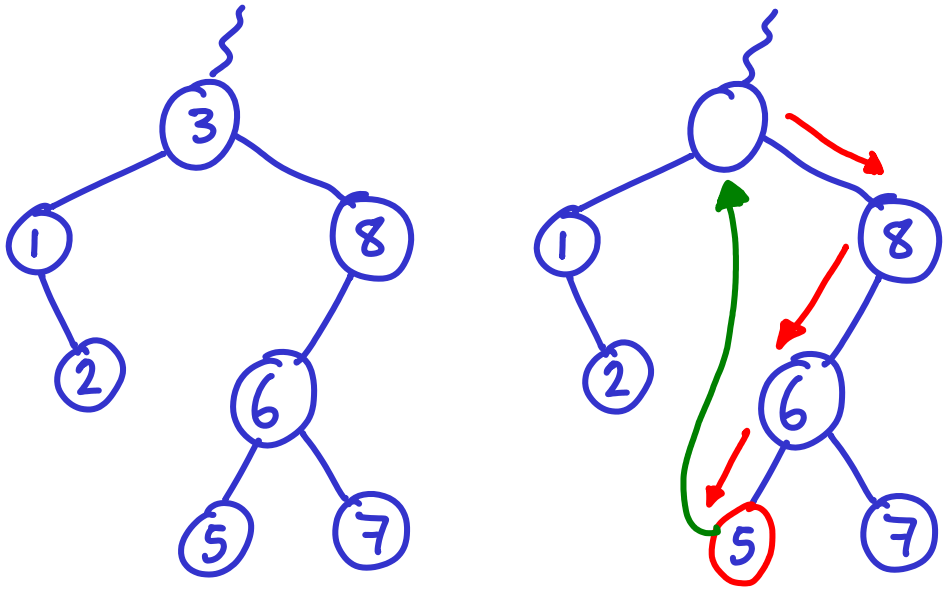
delete(3) → find successor : smallest element greater than 3

(which exists because : 2 children)



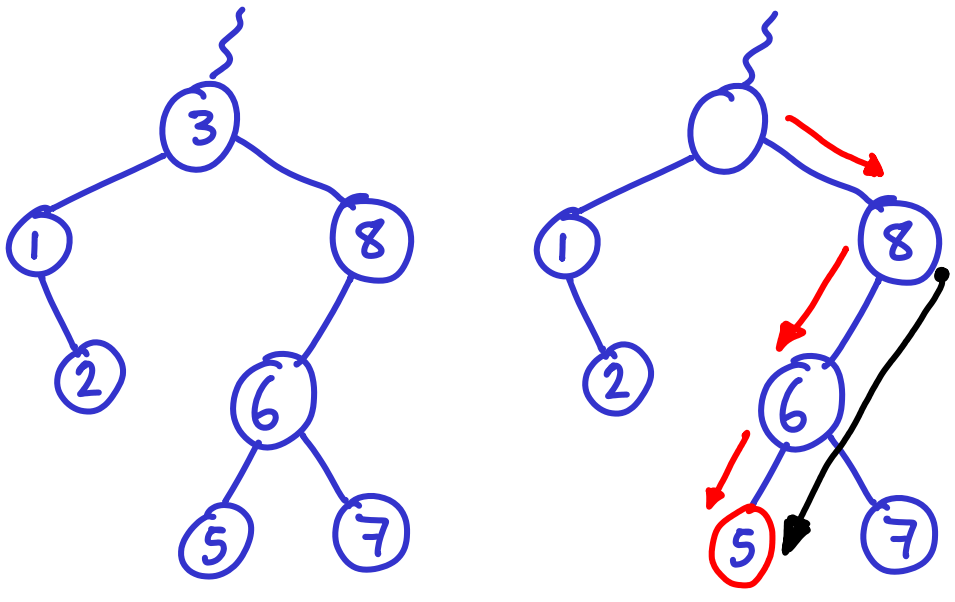
Binary search trees

delete(3) → find successor & replace



Binary search trees

delete(3) → find successor & replace



By definition,

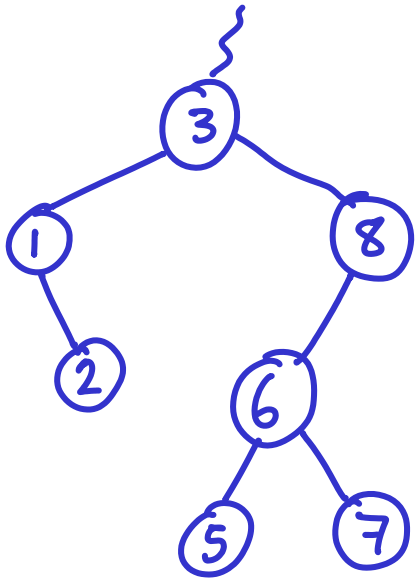
successor is

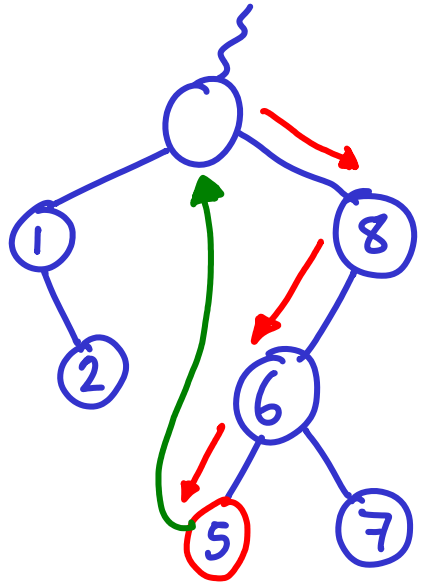
the last node visited on a

↙ path from R-child(3)

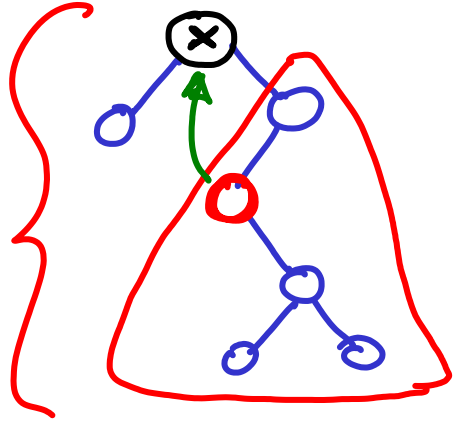
Binary search trees

delete(3) → find successor & replace



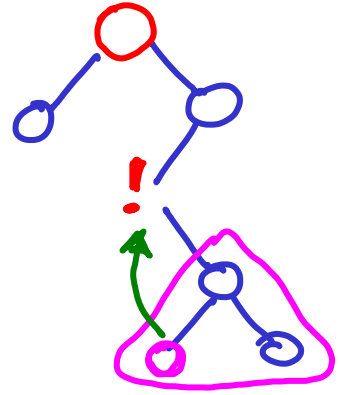


might need to recurse



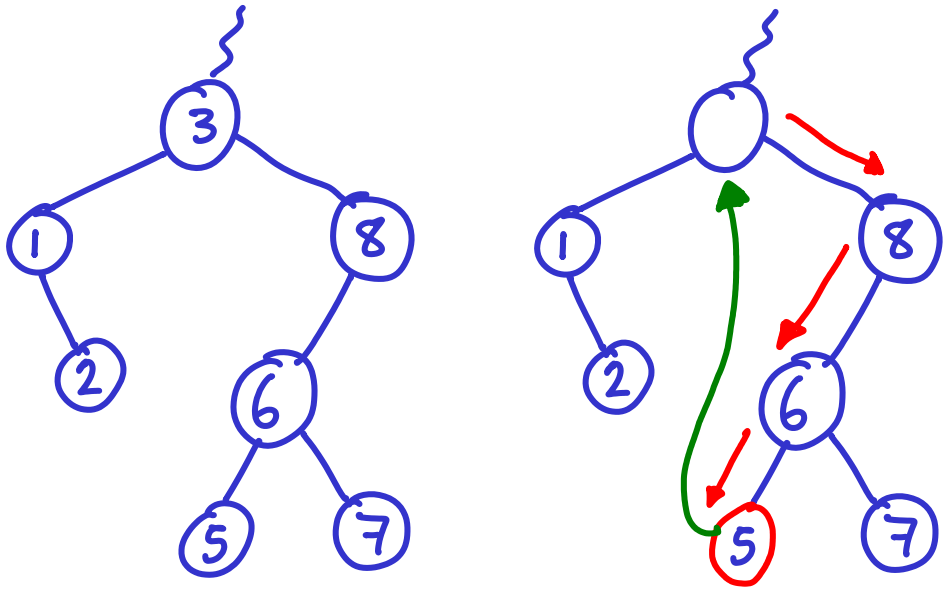
By definition,

successor is the last node visited on a path from R-child(3)



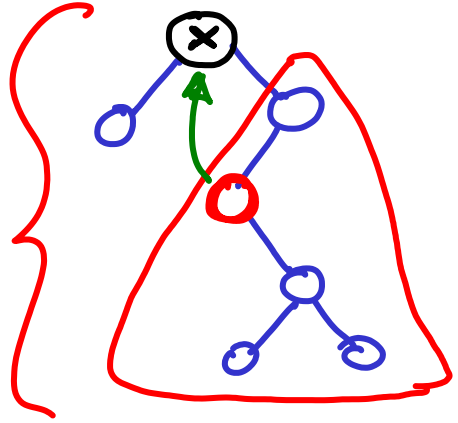
Binary search trees

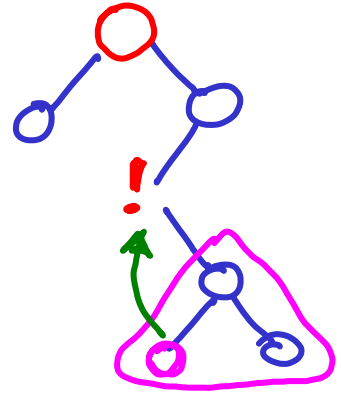
delete(3) → find successor & replace



total O(depth) ← might need to recurse

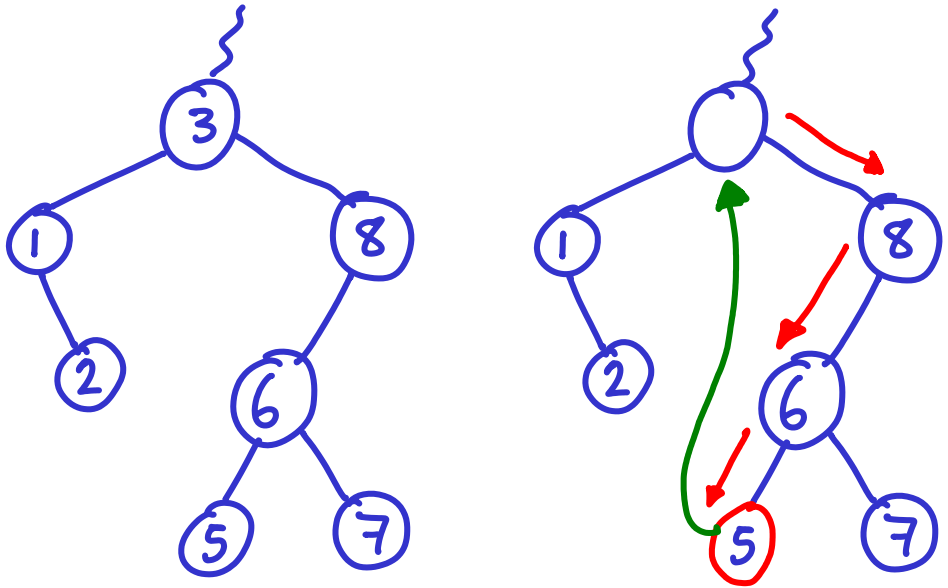
By definition,
successor is
the last node visited on a
path from R-child(3)





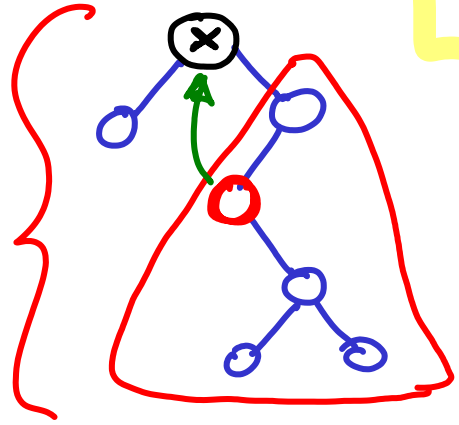
BINARY SEARCH TREES

delete(3) → find successor & replace



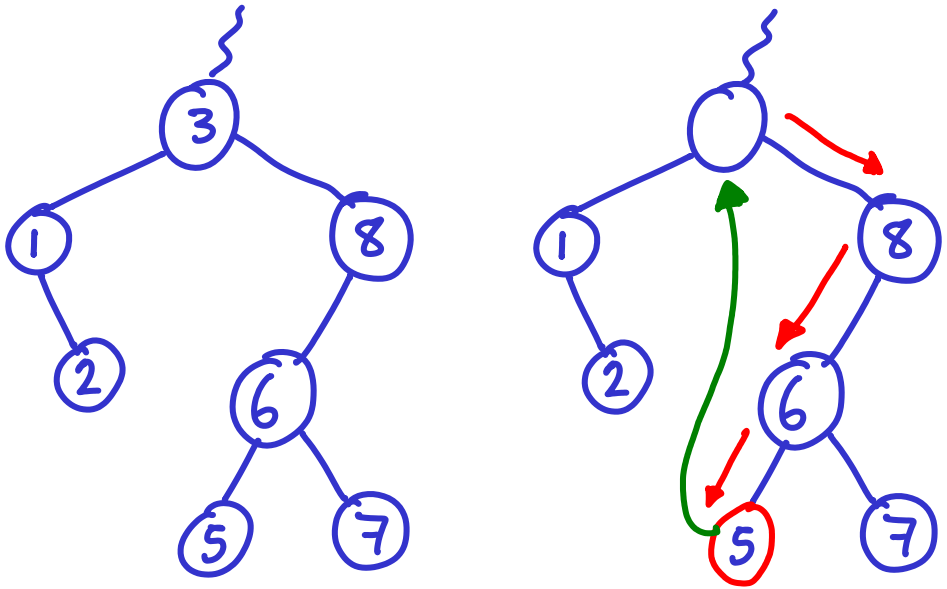
total O(depth) ← might need to recurse

By definition,
successor is
the last node visited on a
path from R-child(3)

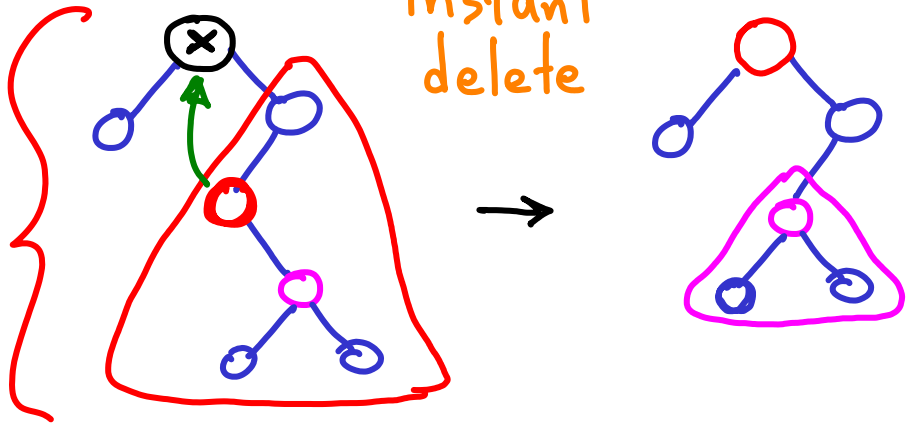


BINARY SEARCH TREES

delete(3) → find successor & replace

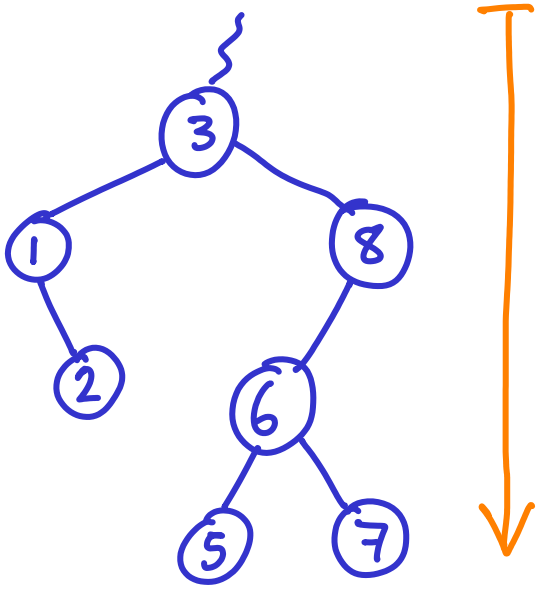


By definition,
successor is
the last node visited on a
path from R-child(3)



still O(depth) ← no need to recurse

Binary search tree summary



INSERT → SEARCH → O(depth)

DELETE → SEARCH → O(depth)

We should keep the tree balanced as much as possible