Intro to Algorithms course

Structure

Asymptotical Analysis
Divide and Conquer
Sorting algorithms (quicksort, mergesort)
Graph Search and Connectivity
BFS DFS
Matching
Randomized Algorithms

Prerequisites for todays lecture:

Runtime analysis, O notation
Sorting algorithms
Basic mathematics (sum of geometric series, logarithms)

Data structures

Data structures organize your Data
in ways you can access them quickly
to produce better programs

e.g. lists, stacks, queues, heaps, search trees, hash tables...

Which data structure should I use?
depends on the type of tasks and operations

e.g. repeated extract min computations: heap

Heap

n objects each object has a key
e.g. objects students records key grade

Operations supported by a heap

INSERT add a new object — Running time: O(logn)

EXTRACT-MIN (or extract-max) — O(logn)

DELETE an object — O(logn)

HEAPIFY create a heap from an array of n objects — O(n)

Application of heaps data structure: Heapsort

Selectionsort: Find the minimum of an array A of elements delete it and put it in the first position of a new array B repeat until A is empty.
Running time: $O(n^2)$

Speed up Selectionsort:
HEAPIFY the array $O(n)$
EXTRACT-MIN of the heap $O(\log n)$ x n times

Total running time: $O(n\log n)$

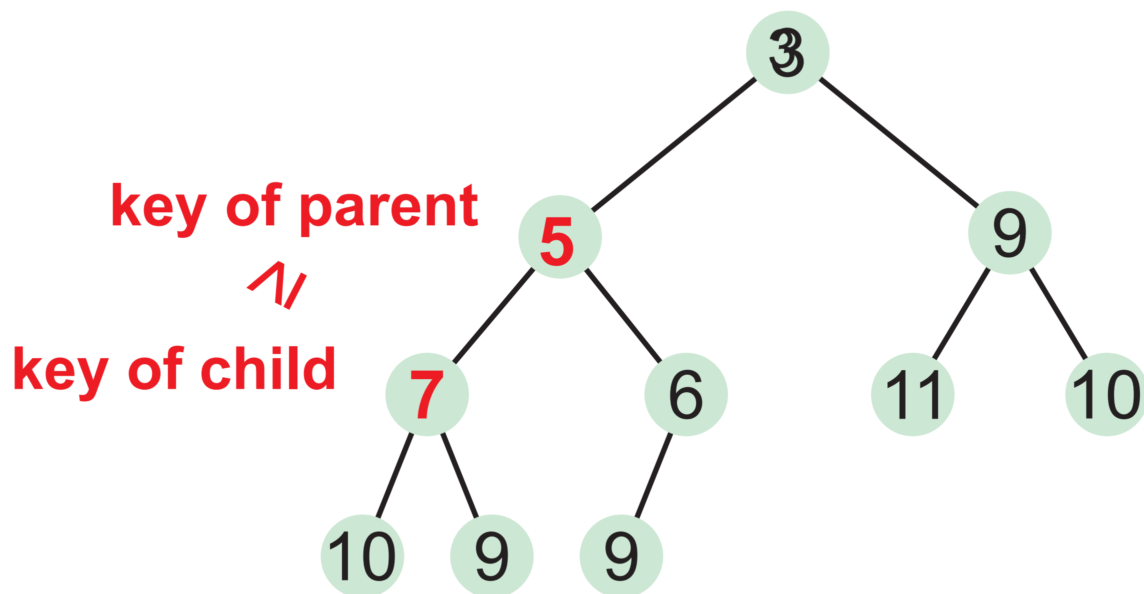
(Binary) Heap visualization

we organize the objects in an almost complete binary tree

Heap property

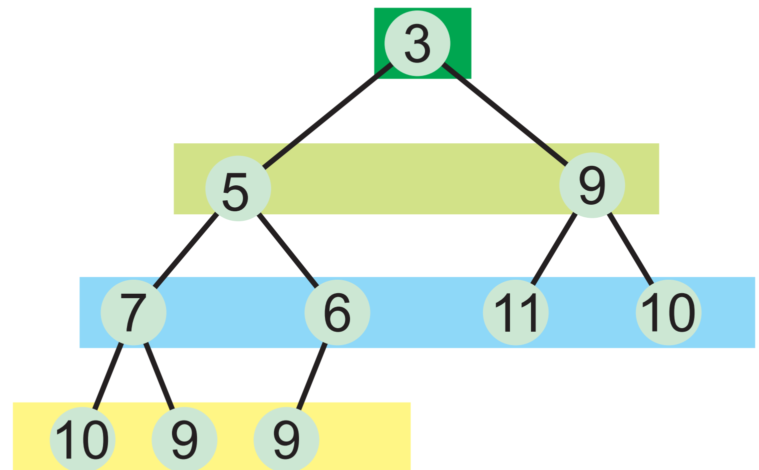
for each object in the heap:

the key in child ≥ key in parent



(Binary) Heap visualization

we organize the objects in an
almost complete binary tree



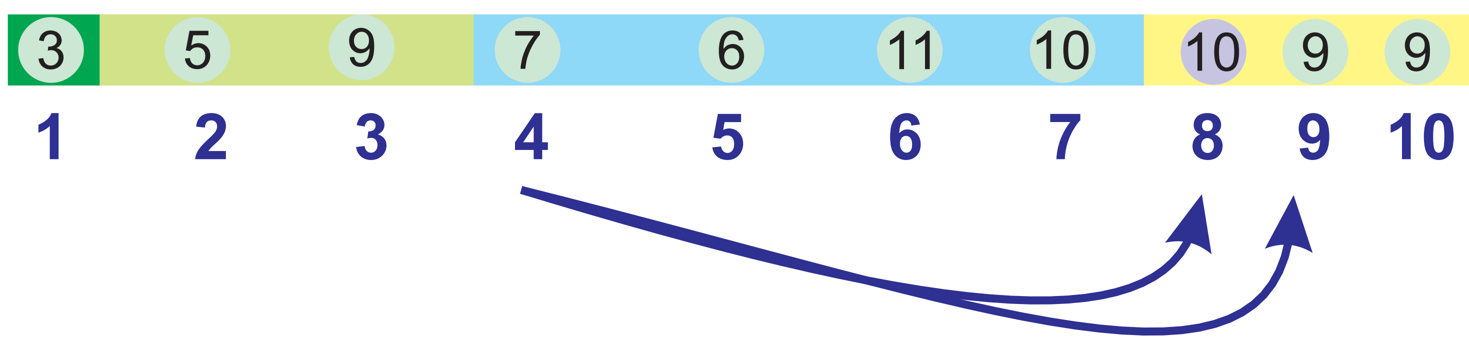
Heap property

for each object in the heap:

the key in child ≥ key in parent

Matrix representation

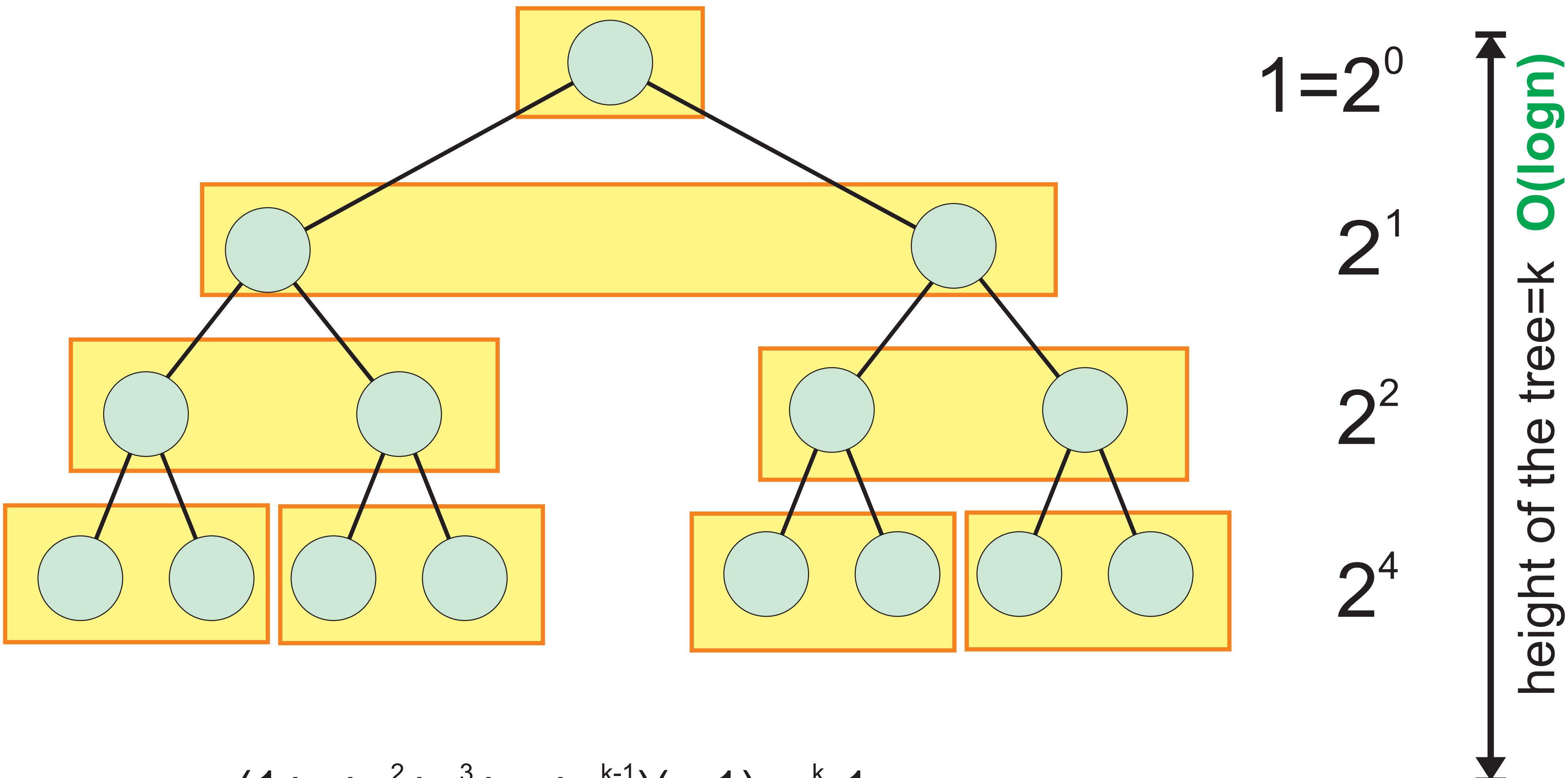
alternative representation as an array



children of i: **2i** **2i+1**

Binary tree

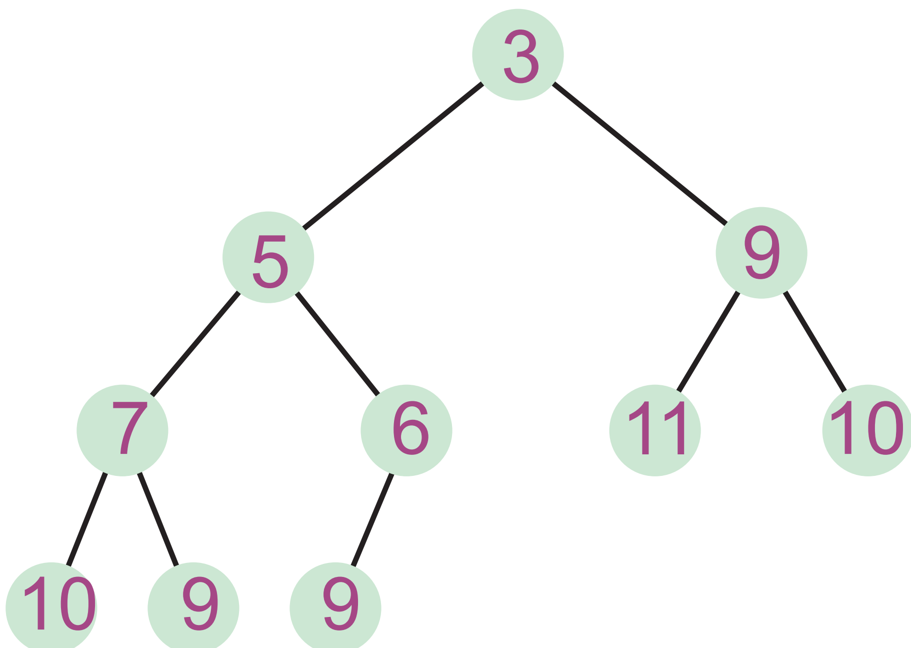
If a complete binary tree has n nodes what is it's height?



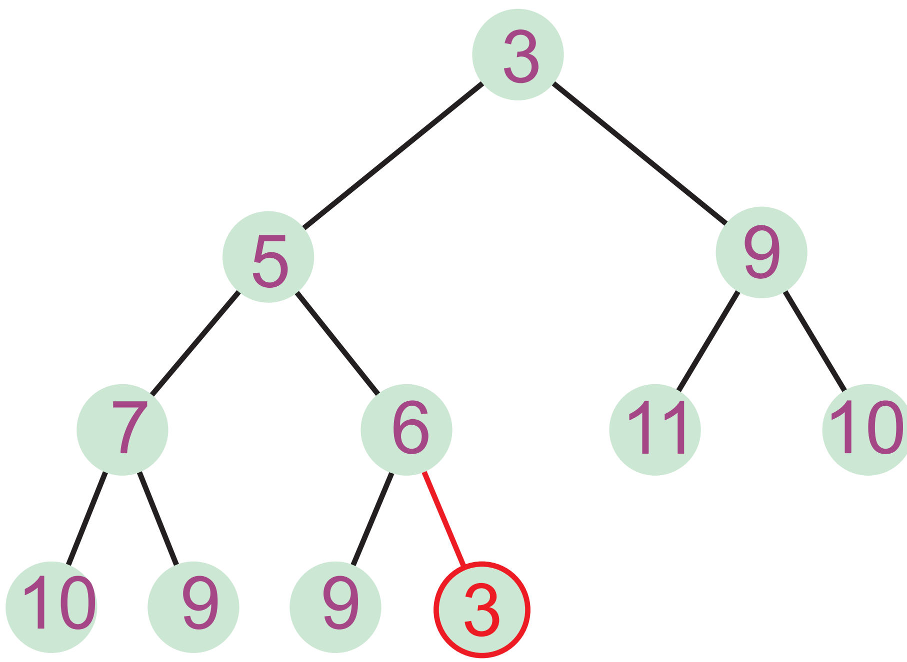
$$(1+x+x^2+x^3+\ldots+x^{k-1})(x-1)=x^k-1$$

$1+2+2^2+2^3+\ldots+2^{k-1}=2^k-1=n$ (number of objects in the heap)

Insert and bubble up operations

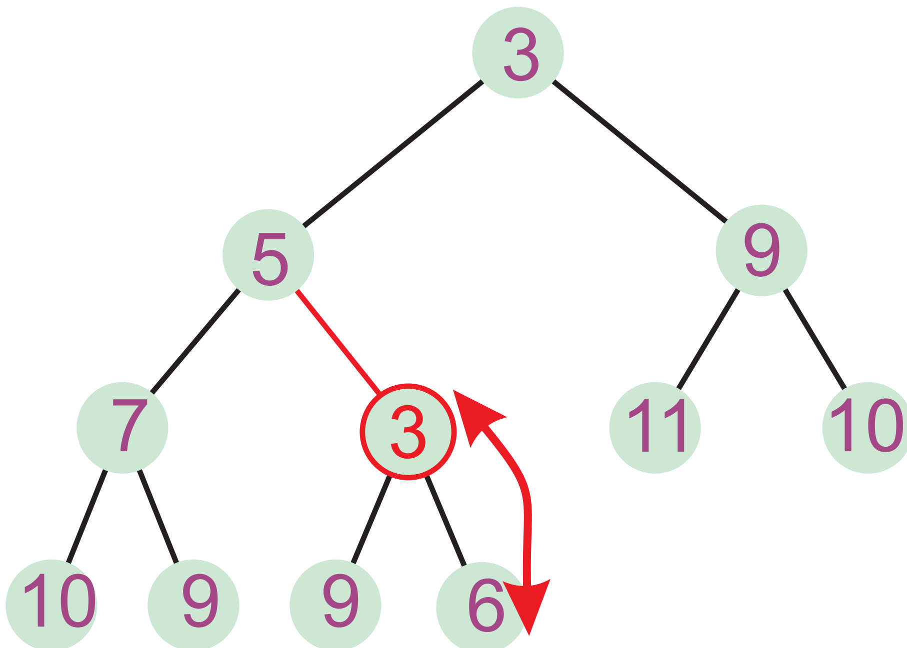


INSERT a new object 3



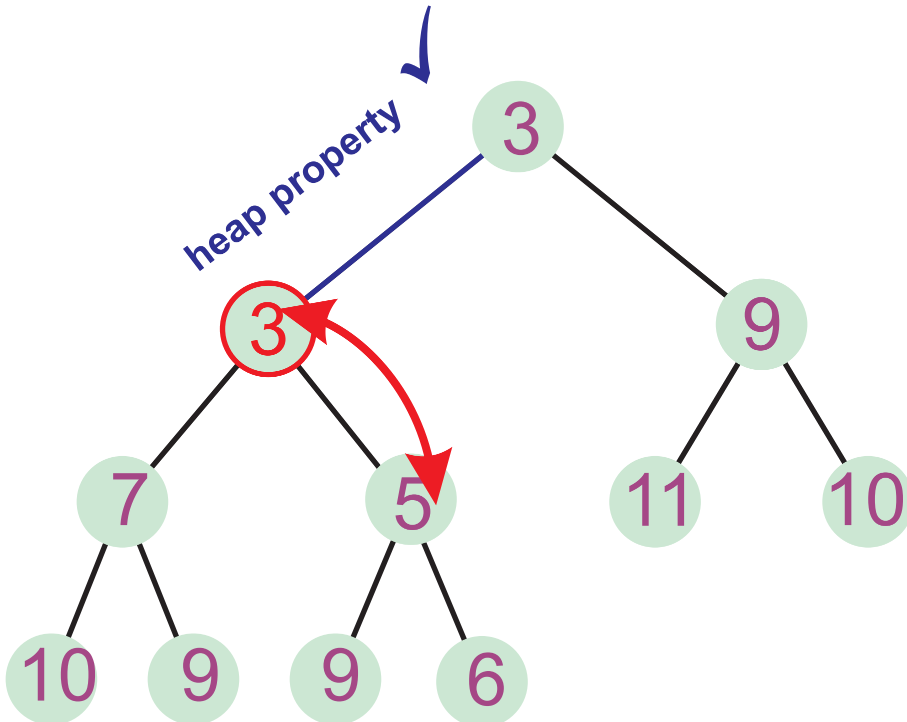
INSERT a new object

make it the last leave
now heap property is violated
bubble-up the new object until heap property is restored

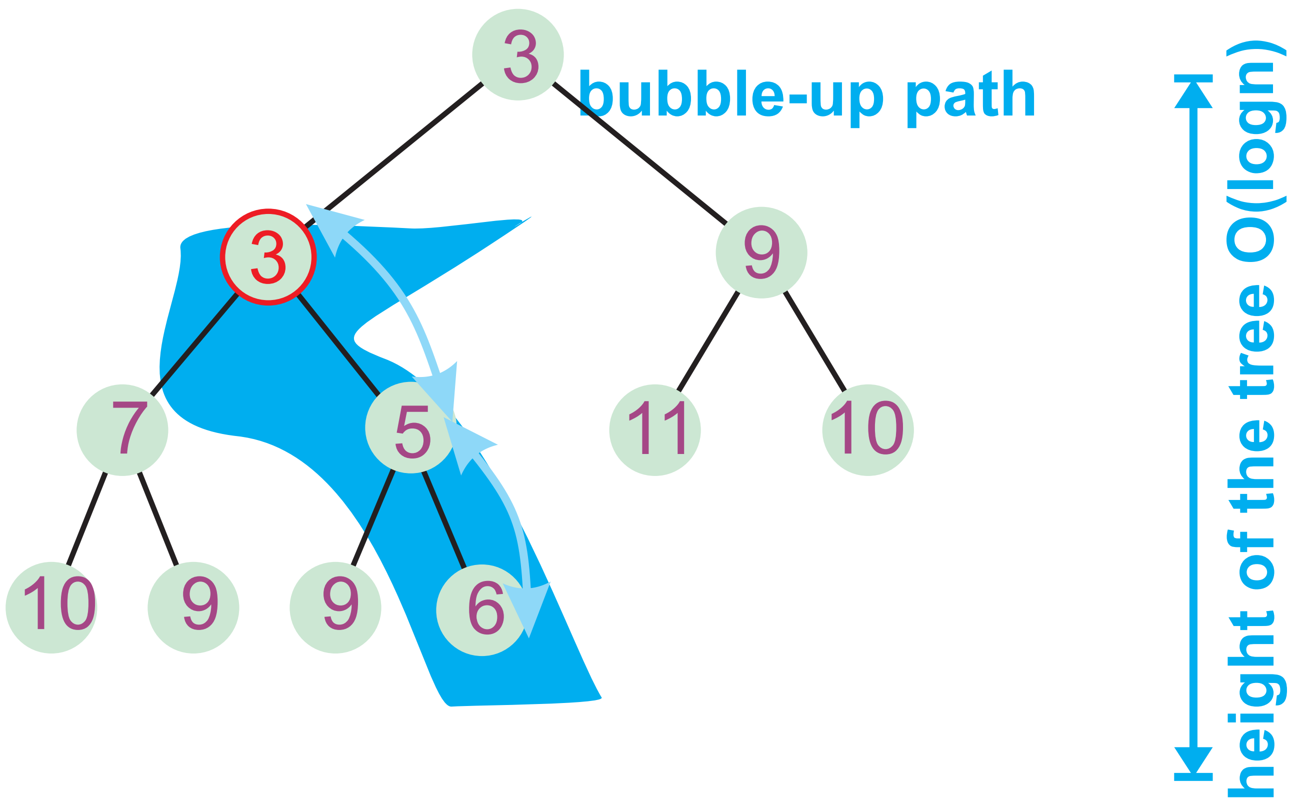


INSERT a new object

make it the last leave
now heap property is violated
bubble-up the new object until heap property is restored

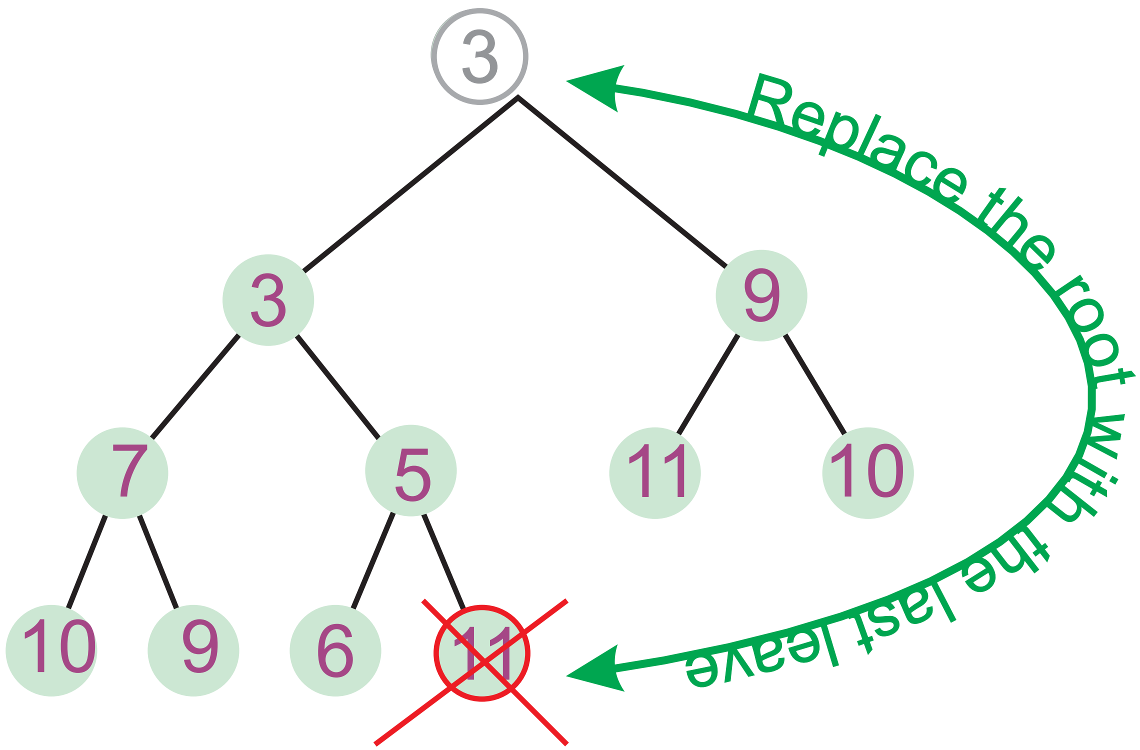


heap property
3
3
9
7
5
11
10
10
9
9
6

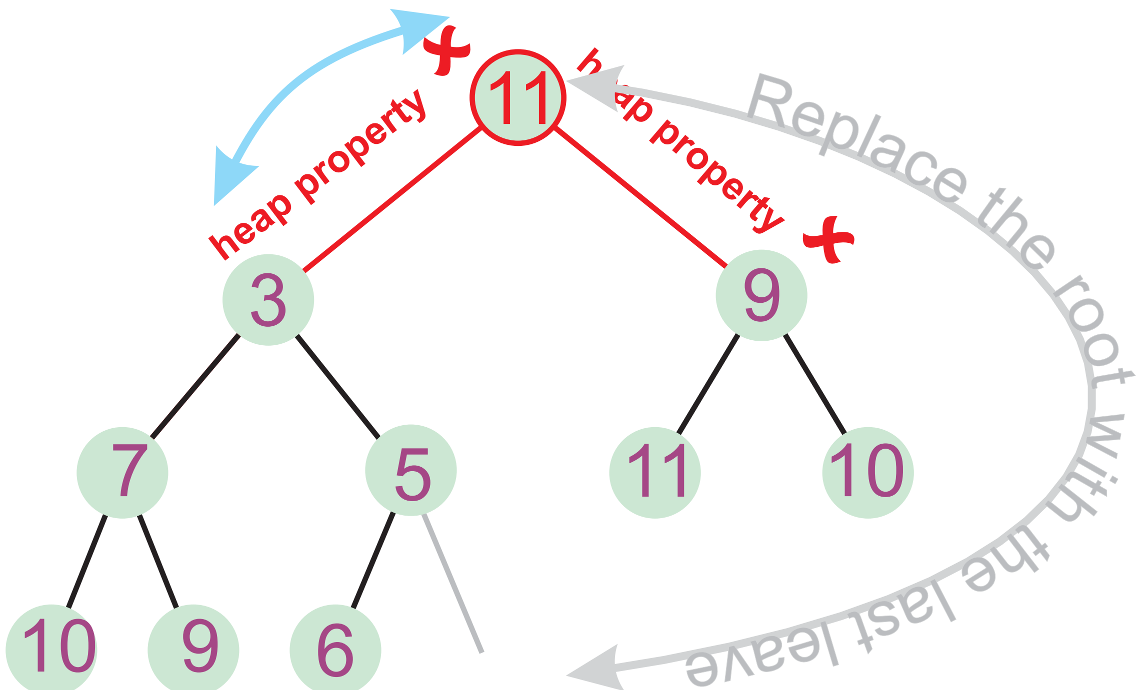


Conclusion: **INSERT** has running time: **O(logn)**

EXTRACT-MIN and bubble down

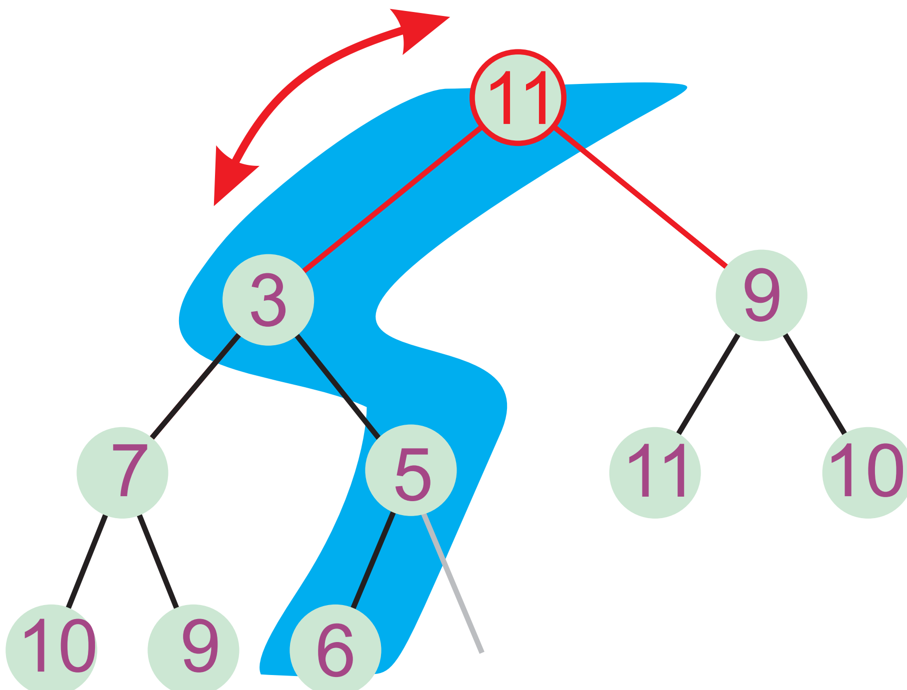


EXTRACT-MIN and bubble down



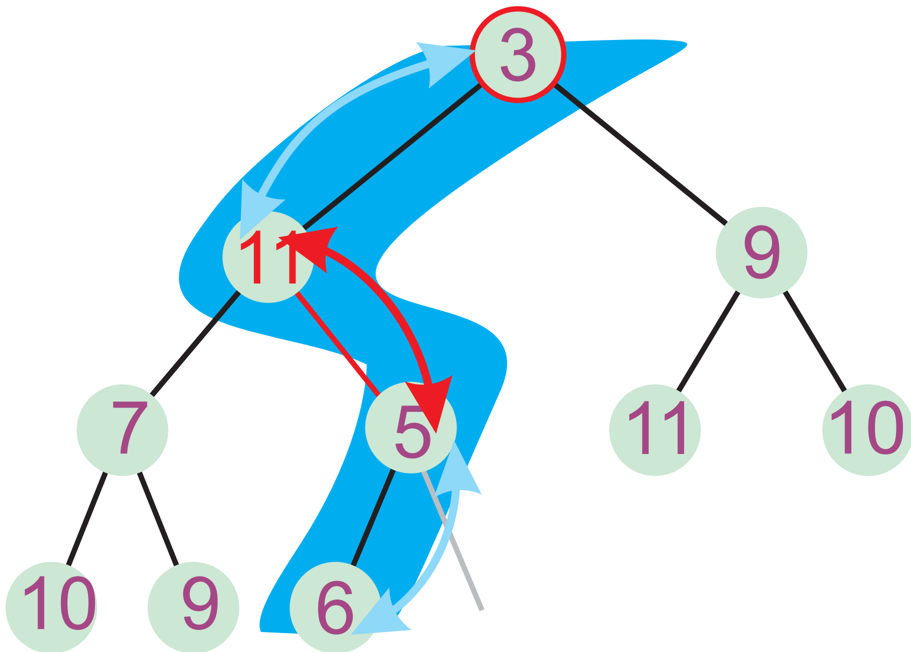
Should we swap with the child
with the smallest or the biggest key?

EXTRACT-MIN and bubble down

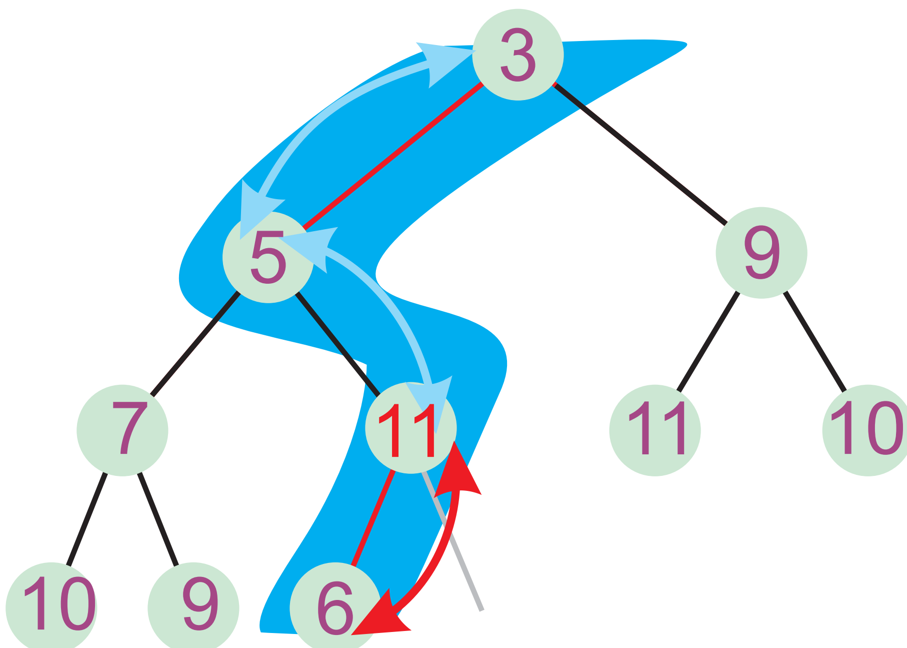


Should we swap with the child
with the smallest or the biggest key?

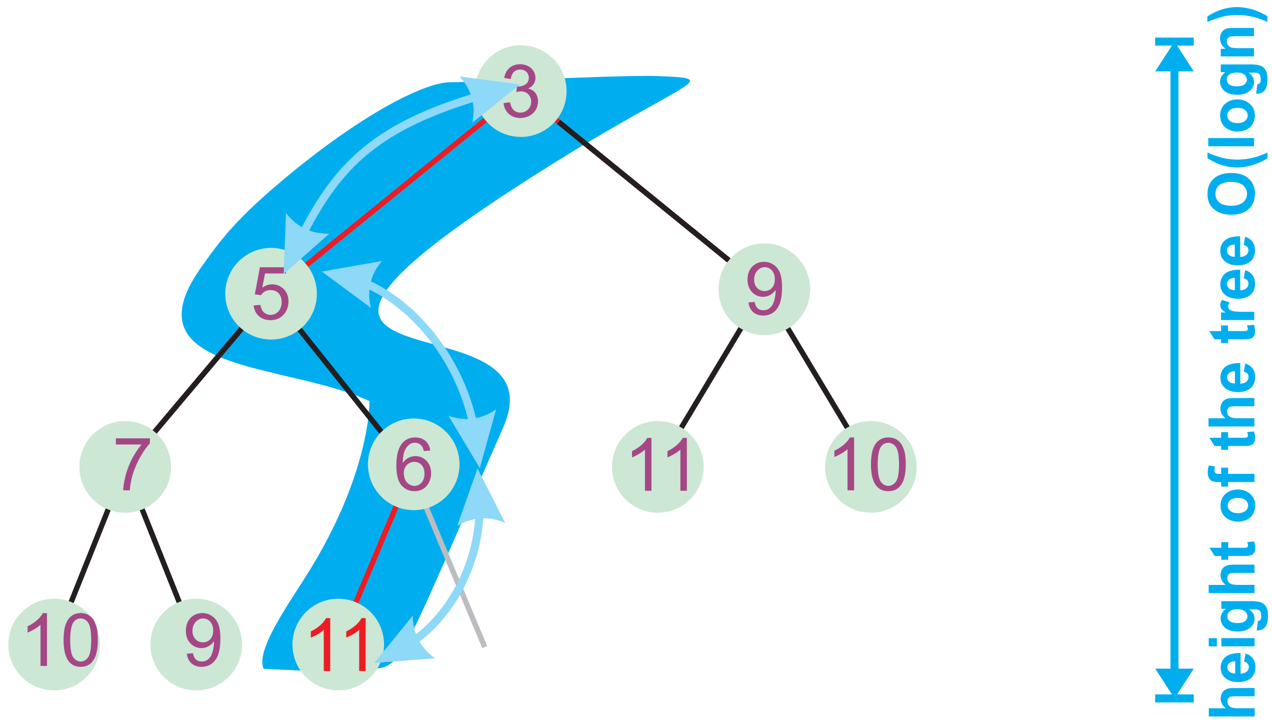
EXTRACT-MIN and bubble down



EXTRACT-MIN and bubble down



EXTRACT-MIN and bubble down



Conclusion: **EXTRACT-MIN** has running time: **O(logn)**