

# ADTs

- \* List
- \* Stack
- \* Queue
- \* Dictionary
- \* Priority Queue

## Priority Queue

void insert (P priority, V value)  
V remove()  
V peek()  
P peek Priority()

What data structures can implement a PQ?

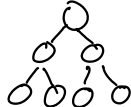
What are their invariants?

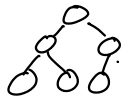
How would insert/remove work?

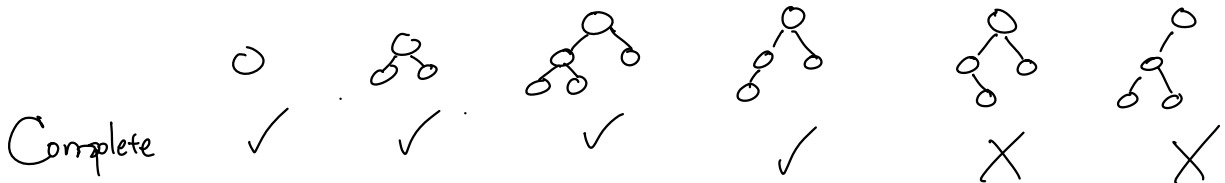
	Balanced BST w/ max ptr	Balanced BST	LL (sorted)	LL (unsorted)
insert	$O(\log n)$	$O(\log n)$	$O(n)$	$O(1)$
remove	$O(\log n)$ or $O(1)$	$O(\log n)$	$O(1)$	$O(n)$
peek	$O(1)$	$O(\log n)$	$O(1)$	$O(n)$

# Max-Heap

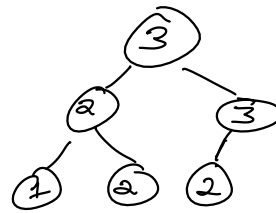
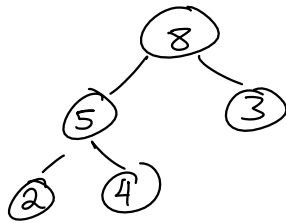
Complete binary tree s.t. every child has value  $\leq$  to its parent

Full:  Every level is completely full or empty.

Complete:  Every level except last is full; last level is packed left



Example heaps:

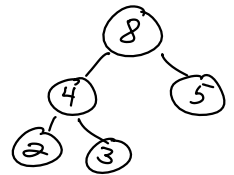


Assume finding correct position for insert is fast (easy to keep complete)

insert(prio, value):

add new node to end of tree w/ (prio, value)

recursively swap w/ parent as necessary: bubbleUp



remove():

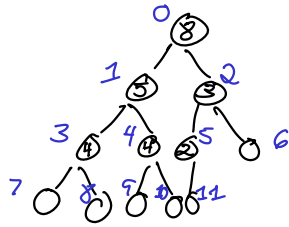
swap root w/ last node

remove node at end of tree

recursively swap w/ largest child as necessary: bubbleDown

# Array Tree

(complete binary trees)



ArrayList

