

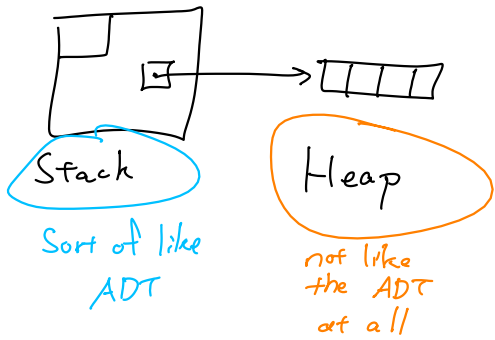
ADT - abstract data type

- List
- Stack
- Queue
- Dictionary
- Priority Queue $\langle P, V \rangle$
 - void enqueue (P prio, V value)
 - V dequeue ()
 - V peek ()
 - P peekPriority ()
 - bool is Empty ()

implementation

Dictionary $\langle P, V \rangle$

(AVL)		reverse
LinkedBST $\langle P, Queue \langle V \rangle \rangle$	Linked List +	main fair sorted
enqueue $O(\log n)$		$O(n)$
dequeue $O(\log n)$		$O(1)$



Binary Tree

all levels except last are packed and last level is packed to the left

every node has priority \geq its children

Complete Binary Tree
is a kind of
Max Heap

is a kind of

is a kind of

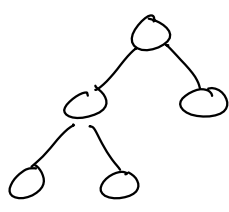
BST

(left desc. less, right desc. greater)

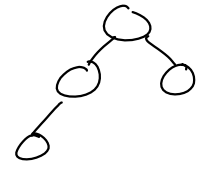
is a kind of

AVL tree

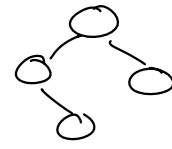
for all nodes, left subtree height & right subtree height differ by ≤ 1



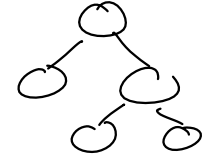
Complete



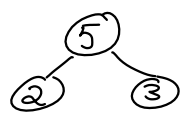
Complete



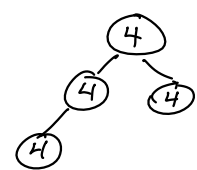
Not Complete



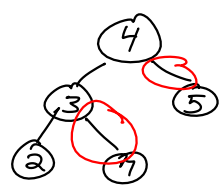
Not Complete



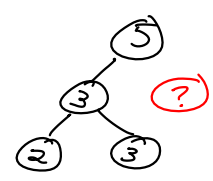
MaxHeap



Max Heap

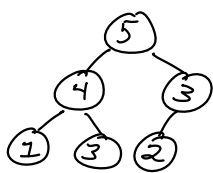


NOT a MaxHeap

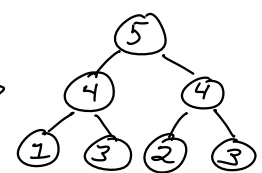


NOT a MaxHeap

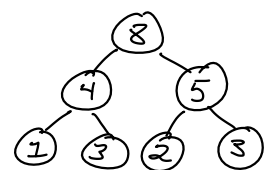
enqueue $O(\log n)$
 add new node to end of tree
 bubble up the new node



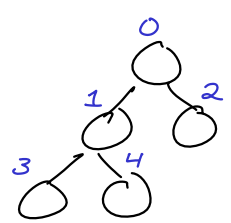
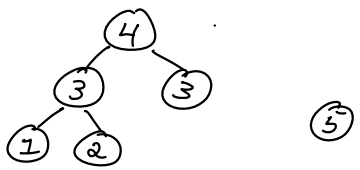
enqueue (4, ~) →



enqueue (8, ~) →



dequeue
 swap root w/ last
 remove last
 bubble down root



complete binary tree
 size 5