Implementing Priority Queues

Priority Queue – a data structure with three key operations:

- see max-priority item
- remove max-priority item
- insert new item

Examples: hospital triage, security threats, Dijsktra's (lowest cost)

How do we build one from scratch? Consider our data structure options so far. How might you use each of these? Which seems best from a performance perspective?

- 'S maintain sorted list, but will pay up to S linear time on insert Linked List Array List
- Tree/Binary Search Tree
- Hashmap/Dictionary what would the key be? Keys be priorities & Values be items with that priority HashSet/set or keys be items mapping to priorities
- Class of our own design (e.g., graph)

What about binary search trees?

Binary Search Tree largest is at bottom right-costs height of the free to get to (12 bottom right If the BST is balanced (bushy rather than tall), it has height of log(n) given n nodes. If we had balanced binary trees, we could insert I delete in log(n) time rather than linear (as with lists) Can we put the max node at root of BST?

here's what 12 BST isn't the right constant we would 10 BST with s' max 2't out + for this problem But, a tree shape could win if We guarantee to keep it balanced. voot _____ oops! lost balance requirement Heaps: a new constraint on binary trees BST: from any node, larger elts all on right & smaller are on the left Heap: a binary-tree (not BST) in which the max item is at the root and both left a right subtrees are also heaps Contrast the highlighted constraintsheap doesn't constrain_left/risht, only the not value of each tree Cnote the long 12 & max "single branch 8 heap (8 is max 1 in this set) 10 tree with 12 at root is also a (not balanced) heep



In both insert & delete, the only items that more are along a single path from root to leaf (sée orange below)









insert 11



Therefore, if we can keep the height of a heap at log (n) for n nodes, we can do insert ldelete in log(n) time & sot-max in constant time Next time, we'll see how to mplement this.

Start of class notes on dynamic programming, buffer cells, on 2D-array creation

