

CS261 Final Review (rough)

Dijkstra's Algorithm.

- assemble $\{\text{visited}\}$, $\{\text{unvisited}\}$

- Set all distances to ∞

for current vertex, examine connected vertices

for each of these, calculate the distance from start to that vertex

update distance

Add to visited, rem from unvisited.

repeat for smallest connecting vertex

AVL Balance

Leaf Nodes: $h=0$

N-L-Nodes: $h = \max(\text{kids}) + 1$

► $\max(\text{kids}) + 1$ | Empty kids = 0

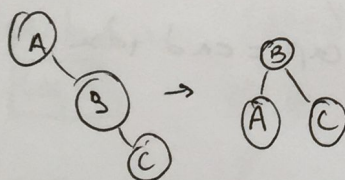
Heavy on:

Balance = right - left

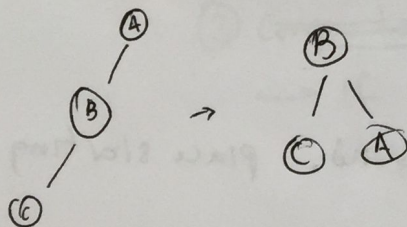
Negatives || Positives
left right

Needs balancing
 $|bal| > 1$

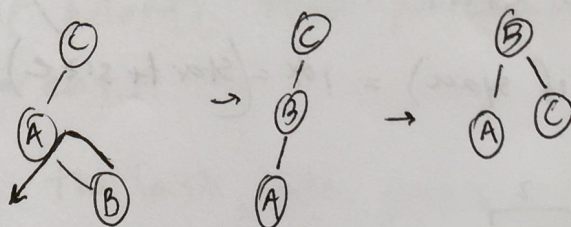
LEFT



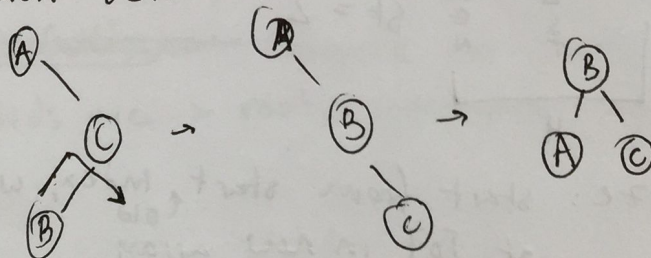
RIGHT



Left RIGHT



RIGHT LEFT



Hash Map

Key : < i.e. name > \rightarrow Hash % table size = index

Value : < data >

Buckets

index [n] = null

if $\left[\begin{array}{c} \text{Steve} \\ 173 \end{array} \right], \left[\begin{array}{c} \text{James} \\ 221 \end{array} \right]$ $\xrightarrow{\text{Hash} \times \text{t.s.} = n}$

index [n] = Steve \rightarrow James \rightarrow null
173 221

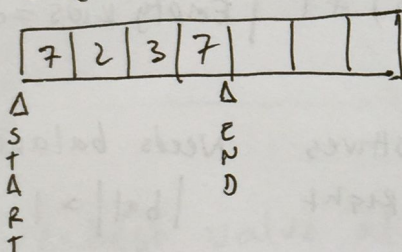
Queue

- FIFO pop from [0], slide all to left. Add to end
insufficient.

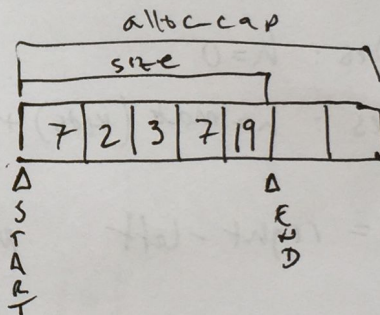
Dequeues

- Add, remove both ends.

Queue (cont'd)



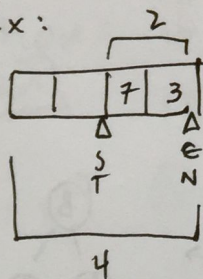
Add ~~19~~
 \rightarrow
End ++



remove : start moves over.

Wrap: (if space) = $\text{idx} = (\text{start} + \text{size}) \% \text{alloc_cap} = \text{end idx}$

Ex:



add '6'
st = 2

$$2 + 2 \% 4 = 0$$

resize: start from start index, wrap around. place starting at [0] in new array.

Dequeues

- use ~~deque~~ queue implementation, but can add/remove both ends.

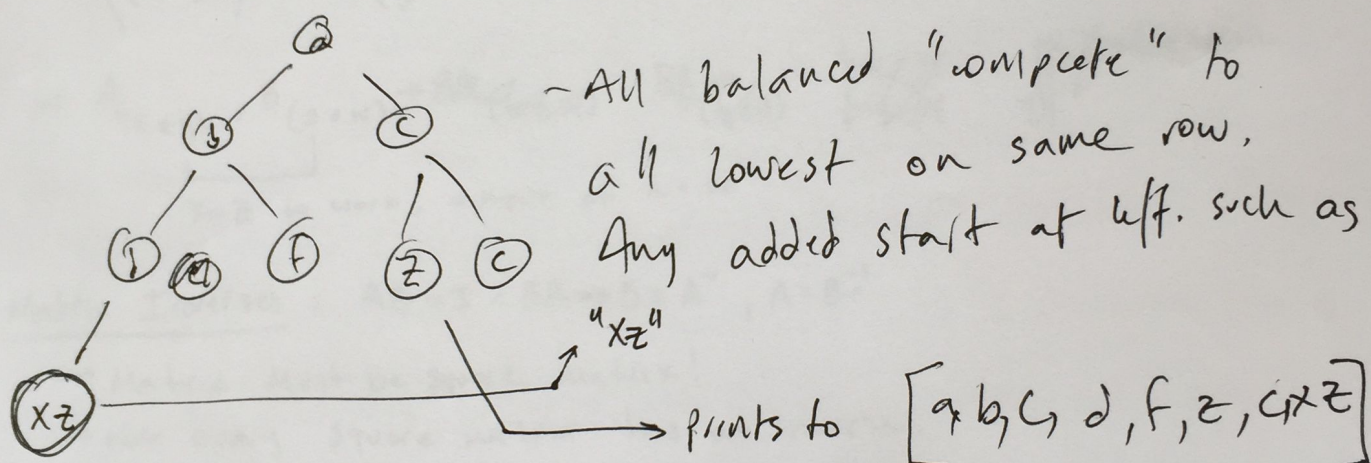
— add loop case for $-1 = \text{start}$ (add start loops end)

~~Priority Queues~~

Heaps

— ADT

- Any node \geq each of its children



- Greatest value always on top [0]

Insertion: Insert where it belongs, then percolate up if it is $>$ parent.

deletion: If we delete the last, No problem (unless)

if we delete the root, ① swap with last leaf.

② ~~comp to left child, and~~ Compare to kids.

→ If both kids are $>$ root, swap with larger child.