CS 252

W, 24 April 2024

Abstract Data Type (ADT)

1. A description of some data
2. A list of operations to be performed on it.

String ADT

needs its own ADT

1. A sequence of characters
2. char at index
   length
   concatenation
   delete char @ index

string comparison
:
:

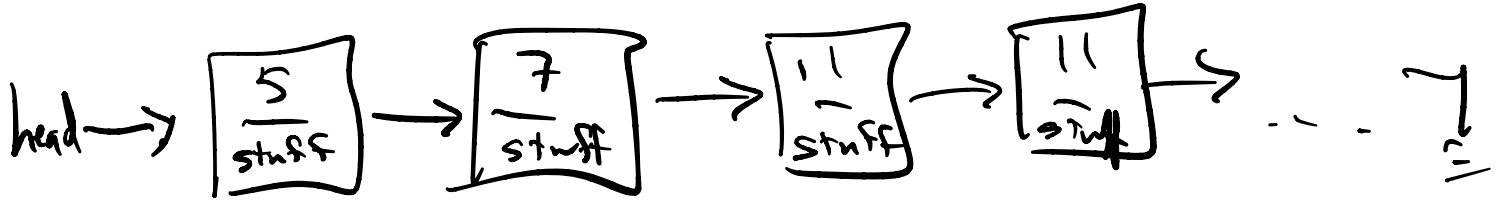
Instances of Strings: "dog", [contents of Hamlet]
:

Priority queue ADT

1. A collection of things, each of which has a "key" from an ordered set
2. Enqueue – add a (key, thing) to the PQ
   Dequeue – remove + return the (key, thing) with the largest (smallest?) key
   Size/length
   ⋮

Dijkstra's Alg needs one of these

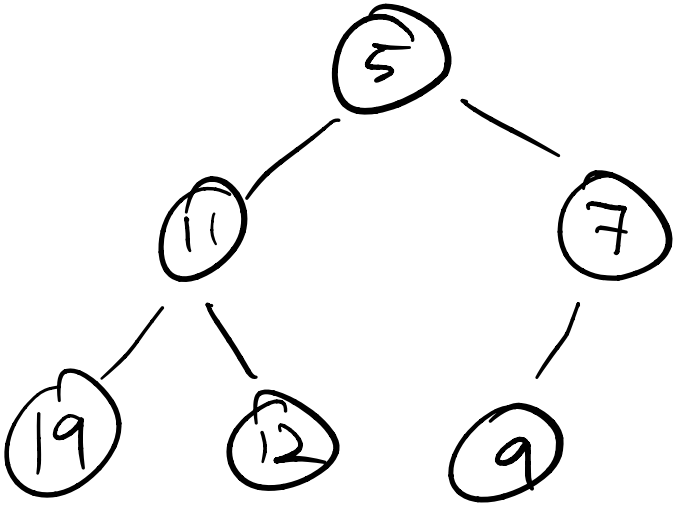
PQ, min, using a singly linked list, sorted by key (min at head)



Enqueue: $O(N)$
(k, s)

Dequeue min-priority: $O(1)$

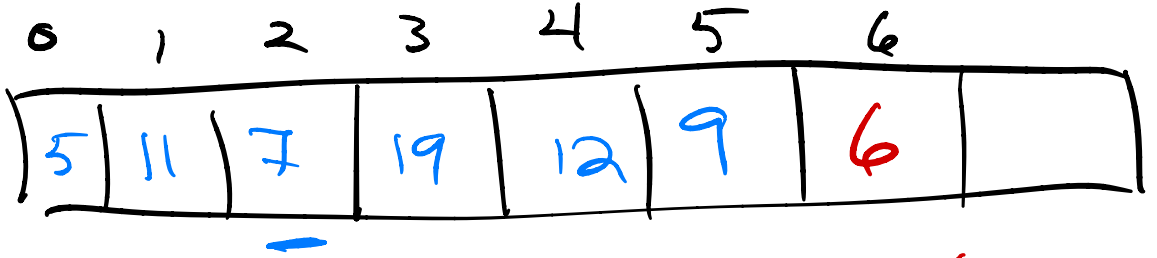
PQ w/ a min-heap



- Binary (rooted) tree
- Each node's key is ≥ its parent's key
- It's "full"
  - each level has maximum # of nodes except bottom level, filled left to right

Enqueue:

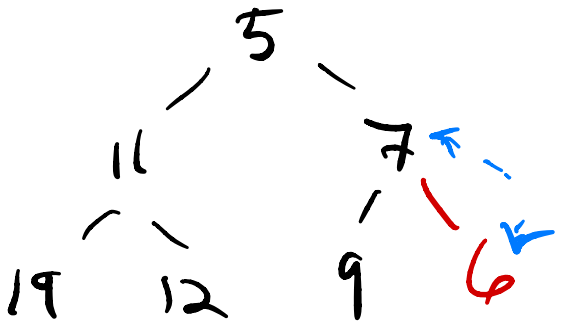
Depends on implementation



Enqueue: O(logN)

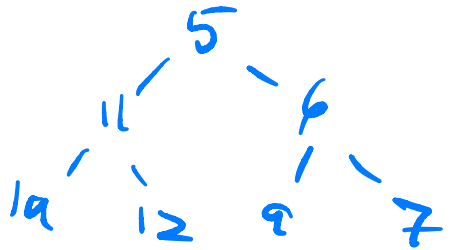
Dequeue: O(logN)

Enqueue: 6



① Stick 6 at the end

② Fix the heap condition by walking the new node up the heap.

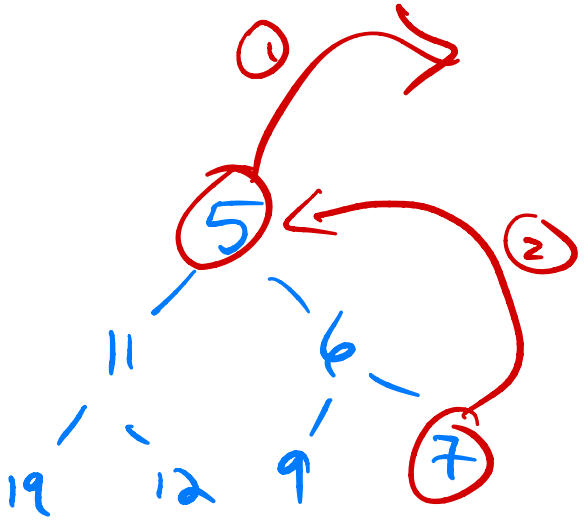


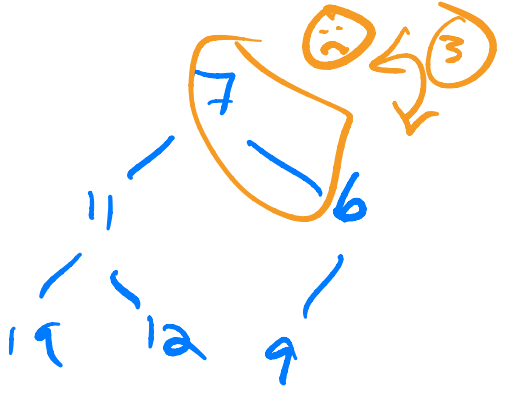
Dequeue: ① remove root

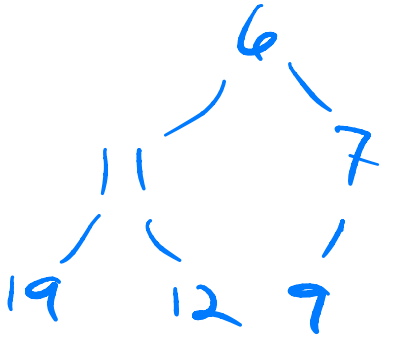
② move bottom right item to the root

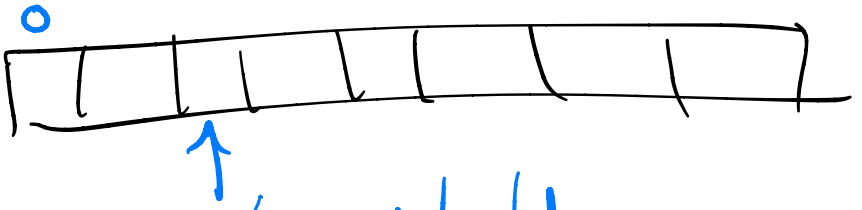
③ fix the heap

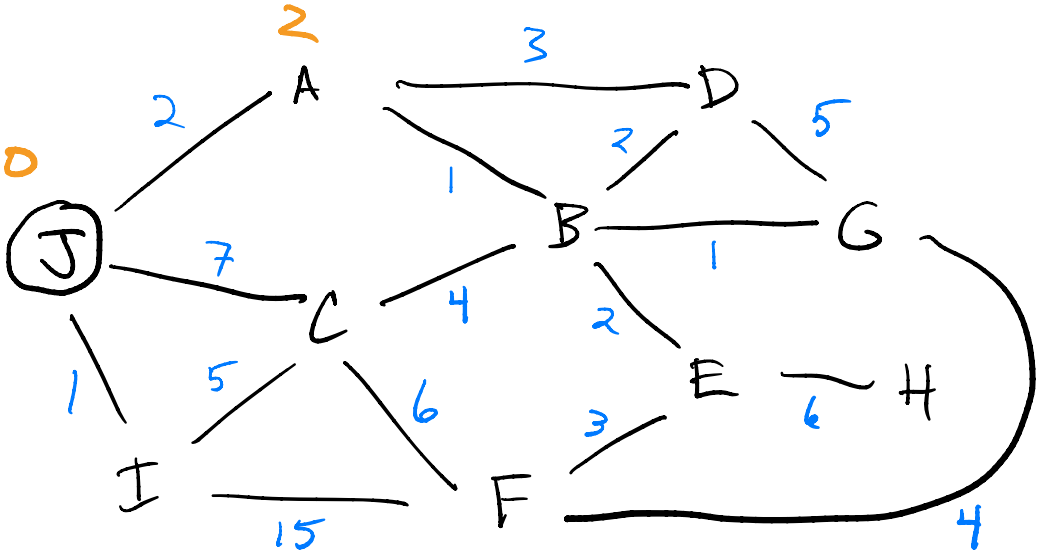
Dequeue
O(logN)








0
k's children
are at indexes
2k+1 & 2k+2



1st iteration

Q: A B C D E F G H I J

dist: ∞ ∞ ∞ ∞ ∞ ∞ ∞ ∞ ∞ 0

2 7 1

u = J

dist[u] = 0

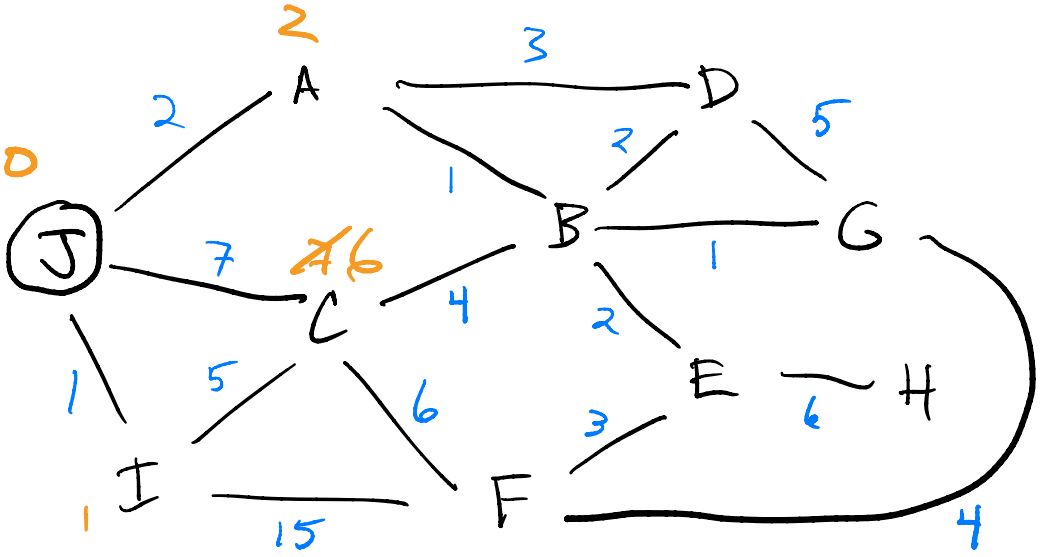
v = A

alt = 0 + 2

2 < ∞ !

if alt < dist[A]

dist[A] = 2



Q: A B C D E F G H I

dist: ∞ ∞ ∞ ∞ ∞ ∞ ∞ ∞ 0

2 7 16 1

6

u = I

v = C

alt = 1 + 5

if alt < dist[u]

dist[u] = alt

6 < 7