5118006-03 Data Structures

Heap

17 May 2024

Shin Hong

Heap

- Heap is a complete binary tree where a consistent ordering exists in every pair of parent and child nodes
  - each element must have a key to represent its priority
  - e.g. the element of a parent node is always greater than or equal to that of its children nodes

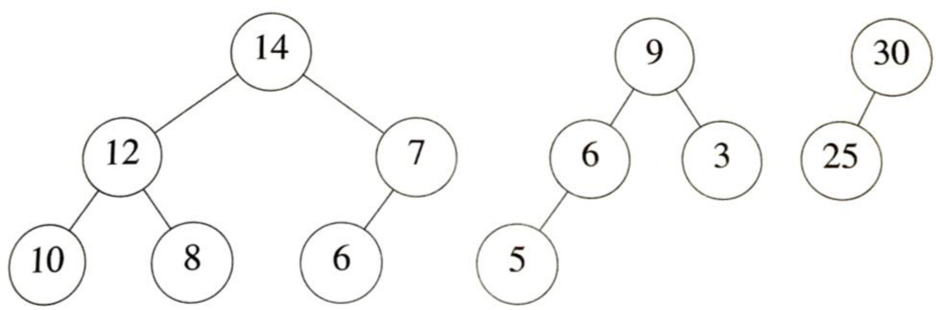
- Heap is frequently used for implementing priority queues

Max Heap

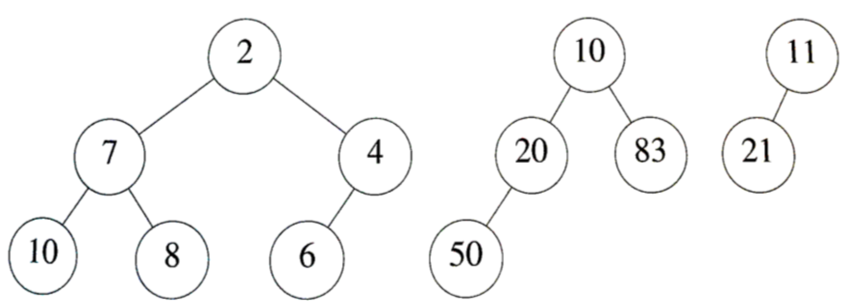
- A max heap is a complete binary tree where the key of a parent is no smaller than the keys of its children
  - c.f., min heap

- Ex.

max heap



min heap



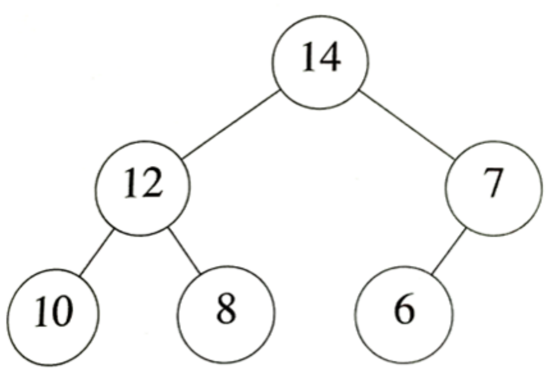
Abstract Data Type

- Objects: an array of elements each of which has a key

- Operations
  - create(M): create a heap of capacity M
  - is_empty(h): check if heap h is empty or not
  - top(h): returns the greatest element in heap h
  - pop(h): remove the greatest element from heap h
  - push(h, e): insert an element e to heap h

Push

- Called insertion or enqueue
- Two requirements
  - keep the binary tree as complete
  - keep the heap property
- Bubble-up algorithm
  1. create the “next” node of the complete tree
  2. place a newly given element to the last node temporary
  3. replace the new node with its parent if they violate the heap property; repeat this until there’s no violation



Push: Algorithm

Input

E [1..*M*], an array of capacity *M* holding *N* elements as a heap
elem, a new element to push in the heap

Output

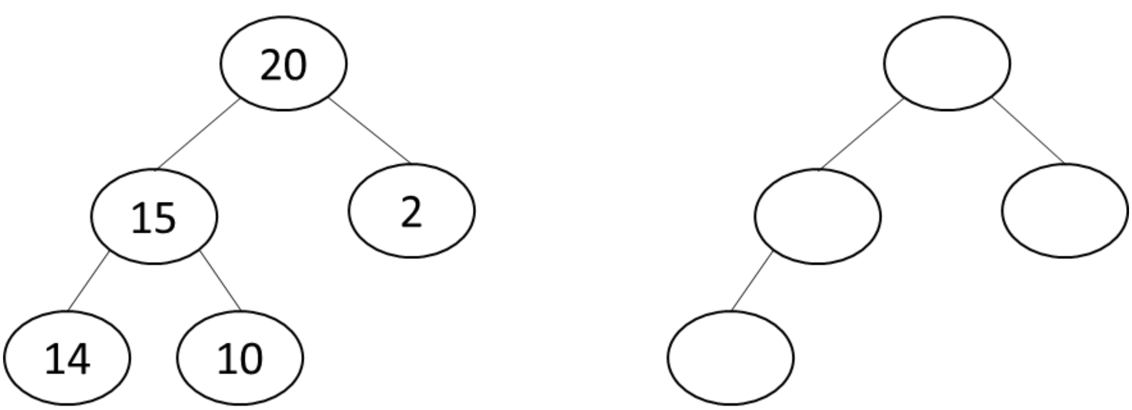
E [1..*M*] holding *N* + 1 elements as a heap

Procedure:

if *N* + 1 > *M* **then** raise an error
N = *N* + 1
E[*N*] = elem
i = *N*
while i > 1 and E[parent(i)] < E[i] **do**
    swap E[parent(i)] and E[i]
    i = parent(i)
end do

Pop

- Called Dequeue
- Algorithm
  1. replace the element in “last” node with that of the root and remove the last node
  2. replace X with the child whose key is greater than its sibling if X violates the heap property; repeat this until there’s no violation
- Example



Heap Sort

- Idea
  - Push all elements to sort to a max heap
  - Pop the greatest one repeatedly until no element remains

- Adjust operation on a heap (i.e., heapify)
  - Assume that a child of the root is already a heap, but the root may not be greater than its children
  - Swap the root node and its greatest child until the heap property is satisfied