Computer Science 331

Binary Heaps

Mike Jacobson

Department of Computer Science
University of Calgary

Lecture #24

Outline

Binary Heaps

Definition: A *binary heap* is

- a binary tree whose nodes store elements of a multiset (possibly including multiple copies of the same value)
- every heap of size *n* has the same *shape*
- values at nodes are arranged in *heap order*

Applications:

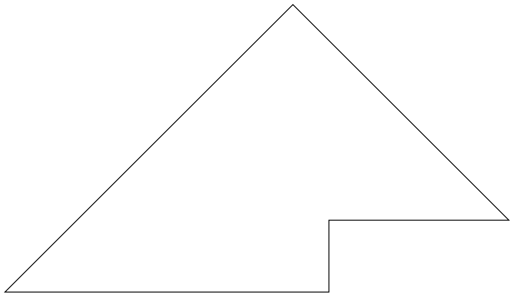
- Used to implement another efficient sorting algorithm (Heap Sort)
- One of the data structures commonly used to implement another useful abstract data type (Priority Queue)

Reference: Text, Section 8.5

Heap Shape

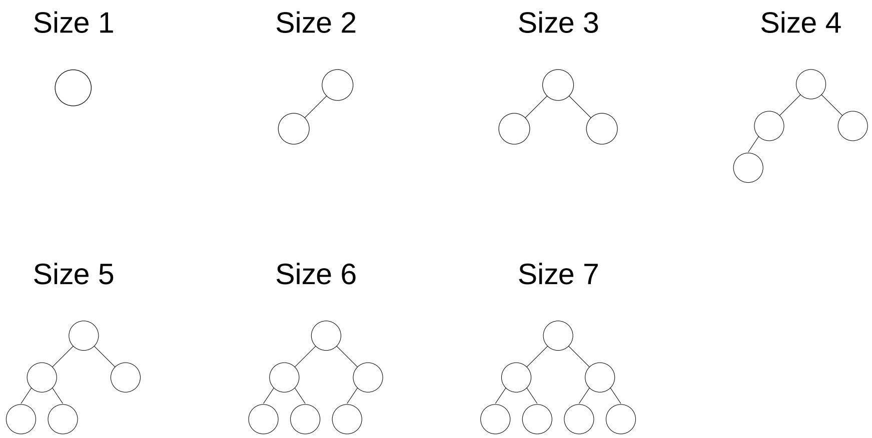
A heap is a *complete* binary tree:

- As the size of a heap increases, nodes are added on each level, from left to right, as long as room at that level is available.



Heap Shape: Examples

Shapes of Heaps with Sizes 1–7:



Height

The *height* of a node, and of a heap, are defined as follows.

- **Height of a Node in a Heap:** Number of edges on the longest path from the node down to a leaf
- **Height of a Heap:** Height of the root of the heap

Theorem 1

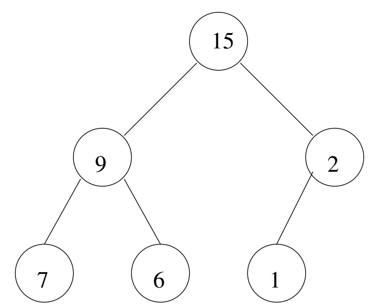
If a heap has size n then its height $h \in \Theta(\log n)$.

Proof: use the fact that a heap is a *complete* tree — every level contains as many nodes as possible.

Proof of Height Bound

Max-Heaps

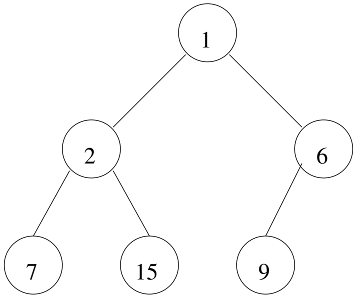
Max-Heaps satisfy the *Max-Heap Property:* The value at each node is *greater than or equal* to values at any children of the node.



Application: The Heap Sort algorithm

Min-Heaps

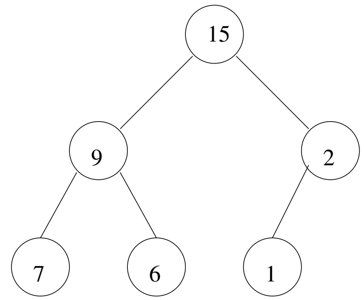
Min-Heaps satisfy the *Min-Heap Property:* The value at each node is *less than or equal* to the values at any children of the node.



Application: Used for Priority Queues

Representation Using an Array

A heap with size n can be represented using an array with size $m \geq n$



0	1	2	3	4	5
15	9	2	7	6	1

Index of Root: 0

For $i \geq 0$

- $\text{parent}(i) = \lfloor (i-1)/2 \rfloor$
- $\text{left}(i) = 2i + 1$
- $\text{right}(i) = 2i + 2$

Representation Using an Array

Suppose A is an array used to represent a binary heap.

Notation:

- $A[i]$: value stored at the node whose index is i
- heap-size(A): size of the heap represented using A
- length(A): size of the array A itself.

Properties:

- $\text{heap-size}(A) \leq \text{length}(A)$
- The entries

$$A[0], A[1], \ldots, A[\text{heap-size}(A) - 1]$$

  are used to store the entries in the heap.

What's Next?

Coming up next. . .

- A “Max-Heapify” operation that ensures the Max-Heap property is satisfied for a subheap with root $A[i]$
- A “BuildHeap” operation that can be reused to produce a heap from a given set of elements (stored as the entries in an array)
- A “deleteMax” operation that can be used to remove the largest element from a heap, generating a slightly smaller heap from the one that was given
- Another efficient sorting algorithm (heapSort) using these components