A primary queue is a collection of items supporting the following operations.

Add – add an item to the collection

Remove-max – remove the highest priority item from the collection

Let’s try to implement a priority queue using a sequence of items.

We can do both operations in O(logn) with AVL trees. But AVL trees are hard to code, and have space overhead (left, right, label). So, it’s kind of overkill.

Let’s look at a new data structure!

A binary heap is a complete or nearly complete binary tree that is heap ordered.

A complete tree has all levels filled in.

A nearly complete tree is complete for all but the lowest level, which is filled in from left to right.

A tree is max-heap ordered if the root of every subtree is the largest value in the subtree

Insertion is a 2-step algorithm

1. Insert a new value to the lowest level which grows from left to right (will not break shape, may break heap order)
2. Repeatedly swap the new value with its parent if it’s larger

Insertion is O(h) but h is O(logn) so implement is O(logn)

Deletion of the root is a 2-step algorithm

1. Replace the root with the right most leaf on the lowest level (will not break shape, may break heap order)
2. Repeatedly swap the new upstart value with the larger of its children as needed.

Complexity: o(logn)

Implementing binary heaps: tree without pointers

Let’s look at a binary heap:

Left-child (i) = 2i + 1

Right-child (i) = 2i + 2

Parent (i) = i-1/2