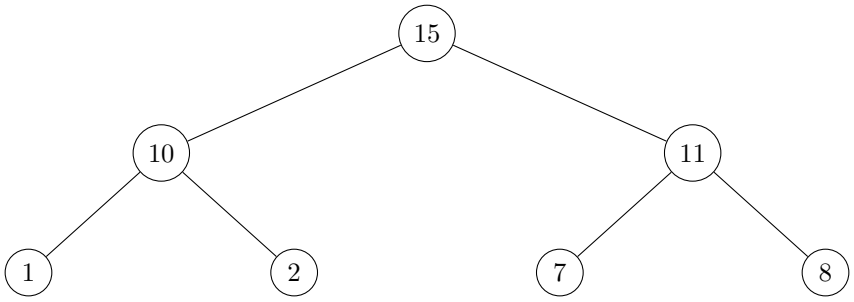
CS 240: Data Structures and Data Management **Fall 2020**

Tutorial 2: September 28

1. Insert 27 and 9 into the following heap, and then perform a delete-max operation on the resulting heap.



2. Given a family k sorted arrays $A_1, \ldots, A_k$, where the combination of the k arrays has n elements, give an $O(n \log k)$ time algorithm that produces a single sorted array containing all n elements. Hint: use a priority queue.

3. Let L denote a sorted array of n distinct integers that are pairwise coprime. Given L and an integer k between 1 and $\frac{n(n-1)}{2}$, write a function that produces a pair (i, j), with $i < j$, such that $\frac{L[i]}{L[j]}$ is the k-th smallest fraction that can be made from elements in L. The algorithm should run in $O(k \log k)$ time.