CS240E W24	**Tutorial 2**	Jan. 22

Overview

- Priority queue implementations (review)
- k smallest sums
- Convexity
- Binary lifting (important technique)
- Binomial heap practice
- Multi-way tree
- Intactive rooks

Problems

Q1. k smallest sums. We are given k arrays, each containing k integers. There are k^k ways to pick exactly one element in each array and calculate the sum of the integers. Give an algorithm to find the k smallest sums among them.

Q2. Convexity.

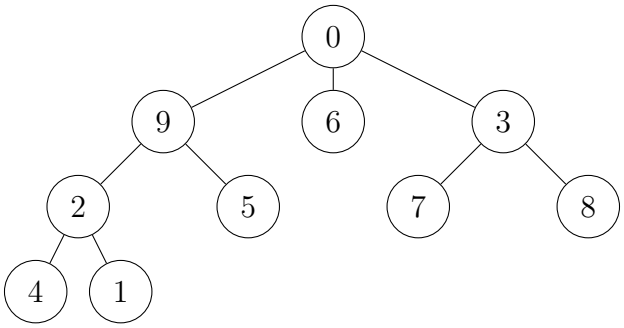
(a) Show that $x \log x$ is a convex function.

(b) Consider the following recurrence relation: $T(0) = 0$,

$$T(n) = n + 1 + \min_{0 \le i \le n-1} \{T(i) + T(n - i - 1)\} \qquad \text{for } n \ge 1.$$

Argue $T(n) \in \Omega(\log n)$ by showing $T(n) \ge (n + 1) \log(n + 1)$.

Q3. Binary lifting. We are given a tree on n nodes (labelled $0, \ldots, n-1$) where node 0 is the root. The tree is represented with the array *parent*, where *parent*$[i]$ denotes the parent of i (and *parent*$[0] = -1$).



	0	1	2	3	4	5	6	7	8	9
parent:	-1	2	9	0	2	9	0	3	3	0

Figure 1: Example tree and corresponding array *parent*.

The *k-th ancestor* of node x in a rooted tree is the node we reach by moving k steps from x towards the root. In Figure 1, the 2-nd ancestor of 1 is 9.

Our goal is to answer *many queries* of the form: given x and k, find the k-th ancestor of x.

(a) Suppose we have a black-box $anc(x, i)$ that returns 2^i-th ancestor of x in constant time.

Give an algorithm to find the k-th ancestor of x in $O(\log k)$ time.

(b) Define the two-dimensional array of integers

$$anc[0..n-1][0..\lceil \lg n \rceil],$$

with the intended meaning:

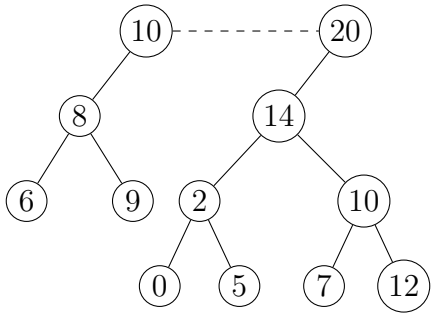
$$anc[x][i] = 2^{i-1}\text{-th ancestor of } x.$$

Suppose that the tree satisfies the min-heap order property. Explain how to compute this array in $O(n \log n)$ time.

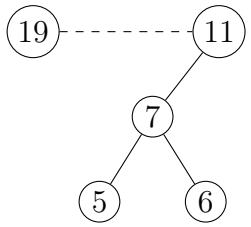
(c) In fact, we sometimes need less than $\log n$ entries in the second dimension of the array *anc*. For a fixed tree T, give an exact tight bound on the size of the second dimension of *anc*.

Additional problems.

Q4. Binomial heap. Perform the following operations on the binomial heap below, in order:



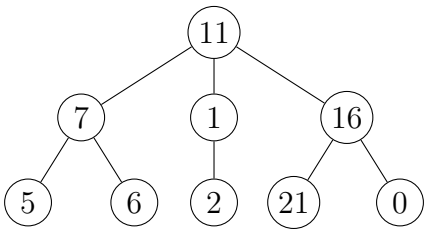
- Insert a node with key 4.
- Perform merge with the following binomial heap:



- Call `deleteMax`.

Q5. Multi-way tree. Let T be a multi-way tree, i.e., nodes can have arbitrarily many children.

(a) There is a simple way to convert a multi-way tree T into a binary tree T': each node of T also becomes a node in T', its leftmost child in T becomes the left child in T', and its sibling to the right in T becomes the right child in T'. Show the binary tree that you get in this way if you start with the following multi-way tree:



(b) For which binary trees T' is there a multiway tree T that it corresponds to? Justify your answer by explaining how you would convert such a binary tree T' into a multiway tree T.

(c) Assume T' is a flagged tree that satisfies the order-property of binomial heaps. What order-property does the corresponding multiway tree T have?

Q6. Interactive rooks. We are given an $n \times n$ chessboard and n rooks. Suppose $n-1$ rooks are placed on the board so that no two rooks attack each other (i.e. no two in the same row or column). We want to place the last rook on the board so that no two rooks attack each other (it can be shown that this is always possible).

However, we cannot see the board. We can only ask questions of the form:

$$question(a, b, c, d),$$

which returns the number of rooks in the region formed bounded between columns a, b and rows c, d (inclusive).

Explain how to determine the placement of the last rook by asking at most $\lceil \log n \rceil$ questions.

Note: We denote the top-left corner has the coordinates $(1, 1)$, and the bottom-right corner has the coordinates (n, n).