CS240E S26	Tutorial 2	May 22

Q1. k smallest sums.

We are given n *sorted* arrays, each containing k integers. Note that there are k^n ways to pick exactly one element in each array and calculate the sum of the integers.

Assume you have access to a set. For a set of size n, there are two available operations: Note: using a set is not required!

- insert with average case time $O(1)$, which adds an element to the set;
- lookup, with average case time $O(1)$, which returns a boolean for whether an element has been previously inserted into the set.

(a) Let $n = 2$. Give an algorithm to find the k smallest sums among these sums.

(b) What about for arbitrary n?

Q2. Convexity.

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

(b) Consider the following recurrence relation:

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

Argue $T(n) \in \Omega(n \log n)$.

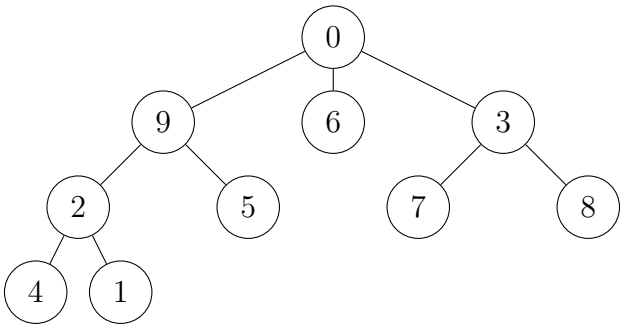
Q3. First i medians. We are given an array a of n integers. For all even $0 \le i < n$, output the median of $a[0..0], a[0..2], \ldots, a[0..2k]$ where $2k \le n - 1$.

Q4. 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*$[0] = -1$).

The *kth 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 second ancestor of 1 is 9.

Our goal is to answer queries of the form: given x and k, find the kth ancestor of x.

(a) Suppose we have access to the function $anc(x, i)$ that returns 2^ith ancestor of x in constant time. Give an algorithm to find the kth ancestor of x in $O(\log k)$ time.



	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*.

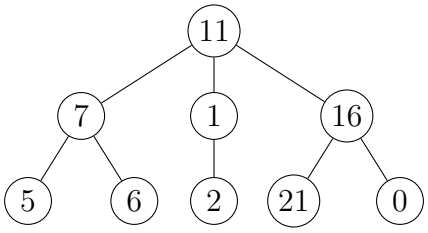
(b) Define the array of integers $anc[0..n-1][0..\lceil \lg n \rceil]$, with the intended meaning:

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

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

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?