CS2040S: Data Structures and Algorithms

Exercises/Optional Problems for Week 2

For: February 10, 2020

Problem 1. Online Median Finding

Given an array of elements in increasing order $a_1, a_2, \ldots, a_n$, we define the median to be the $(\frac{n}{2}+1)^{th}$ element if n was even and the $\frac{n+1}{2}^{th}$ element if n was odd.

Now given an unsorted array of integers, finding the median element can actually be done in expected time $O(n)$[1]. But what if the numbers came in one by one (and not guaranteed to be in increasing order) and you were sometimes expected to find the median halfway through, and remove it?

For example, imagine that the first few numbers that came in were $4, 2, 3$, at which point the median would be 3. Say we had to remove it after that, now we're left with $4, 2$. Then say the numbers $8, 2, 7, 1$ also came in after we removed the median, so all in all now our array looks like $4, 2, 8, 2, 7, 1$, and then we wanted to find and remove the median again, then it would be 4. After that it we were to remove the median again, it would be 2, and so on.

Your job is to is to make use of data structures to support the following functions:

- `void insert(int x)` Given a a value x, insert it into your data structure in $O(\log n)$ time.
- `int getMedian()` Returns the median element of all the elements seen so far and removes it in $O(\log n)$ time.

As one more example, consider the following sequence: $4, 2, 3, 1$, followed by 4 `getMedian()` operations: then you should expect to see $3, 2, 4, 1$.

[1]if you're lucky, you'll also see that in CS3230 it can also be done in worst case $O(n)$.