Linear-time median finding

9/20/24

Administrivia

- HW 2 (AVL trees, divide & conquer) due Tuesday night

- Reading
  - For Monday: Chapter 26 intro and Section 26.1 part 1 (pp. 748-759; up to "scheduling")
  - For Wednesday: Finish Section 26.1

Order statistics

- Given an unsorted array, find the k^{th} smallest element

Order statistics

- Given an unsorted array, find the k^{th} smallest element

How long does it take to find the smallest value (k=1)?

A. $\Theta(1)$ (pretend these are thetas…)
B. $\Theta(\log n)$
C. $\Theta(n)$
D. $\Theta(n \log n)$
E. None of the above

Order statistics

- Given an unsorted array, find the k^{th} smallest element

How long does it take to find the smallest value (k=1)?

A. $\Theta(1)$ (pretend these are thetas…)
B. $\Theta(\log n)$
C. $\Theta(n)$
D. $\Theta(n \log n)$
E. None of the above

Finding the k^{th} smallest: One approach

- Keep a sorted array of the k smallest *so far*
- Traverse the input array and update this preliminary answer

Finding the k^{th} smallest: One approach

- Keep a sorted array of the k smallest *so far*
- Traverse the input array and update this preliminary answer

How long does this algorithm take?

A. $\Theta(n)$
B. $\Theta(n+k)$
C. $\Theta(n \log k)$
D. $\Theta(nk)$
E. None of the above

Finding the k^{th} smallest: One approach

- Keep a sorted array of the k smallest *so far*
- Traverse the input array and update this preliminary answer

How long does this algorithm take?

A. $\Theta(n)$
B. $\Theta(n+k)$
C. $\Theta(n \log k)$
D. $\Theta(nk)$
E. None of the above

Finding the k^{th} smallest: Other approach

- Sort the input array
- Return array[k-1]

Finding the k^{th} smallest: Other approach

- Sort the input array
- Return array[k-1]

- Slow for large and small k, better for intermediate values
- Median: $n/2^{th}$ smallest

Linear-time Select(k)

1. Divide array into ⌊n/5⌋ groups of 5 elements
2. Sort each group of 5 and identify median of each group
3. Recursively find median of these medians
4. Partition array around median of medians (same code as quicksort)
5. Let i be position of the median of medians x

   If i=k, return x

   If i < k, return Select(k-i) on larger part

   If i > k, return Select(k) on smaller part

Practice problem 4

An *inversion* in a sequence is a pair of values such that the larger value is before the smaller one. For example, the sequence {3,30,18,25} has 2 inversions because 30 should be after 18 and 25.

You can modify insertion sort to count the inversions in an array while sorting it. Recall that insertion sort swaps adjacent values when they are out of order. Each swap removes 1 inversion so counting swaps also counts inversions. Unfortunately, this algorithm takes $O(n^2)$ time.

Develop an algorithm for this problem that runs in O(n log n) time. (Hint: Start with mergesort.)

Practice problem 5

https://leetcode.com/problems/k-closest-points-to-origin/

Given n unordered points in the plane, find the k points closest to (0,0).
The distance between (x_1,y_1) and (x_2,y_2) is $\sqrt{(x_1 - x_2)^2+(y_1 - y_2)^2}$.