CISC 621, Assignment 2, Question 5a-c

Graded by: Group 5 (Li Chen, Samuel Kalet, Jing Wu)
Adopted from a solution by: (a) Group 6 (b) Group 6 (c) Group 5

Part A

Problem Statement

Argue that the median of $x_1, x_2, \ldots, x_n$ is the weighted median of the x_i with weights $w_i = \frac{1}{n}$ for $i = 1, \ldots, n$.

Solution

The median of n points is the i^{th} element in the set, where $i = \lfloor (n+1)/2 \rfloor$. This set is assumed to be sorted.

Assume x_k is a median. In order for x_k to satisfy the definition of a weighted median it must be true that:

$$\sum_{i<k} w_i \leq \frac{1}{2} \tag{1}$$

$$= \sum_{i<k} \frac{1}{n} \tag{2}$$

$$= \frac{1}{n}(\lfloor \frac{n+1}{2} - 1 \rfloor) \tag{3}$$

$$= \begin{cases} \text{if n is odd, } \frac{1}{n}(\frac{n+1}{2} - 1) = \frac{1}{n}(\frac{n+1-2}{2}) = \frac{n-1}{2n} \leq \frac{1}{2} \\ \text{if n is even, } \frac{1}{n}(\frac{n}{2} - 1) = \frac{1}{n}(\frac{n-2}{2}) = \frac{n-2}{2n} \leq \frac{1}{2} \end{cases} \tag{4}$$

Since x_k has been shown to always be the weighted median, it follows that the median of $x_1, x_2, \ldots, x_n$ is also the weighted median of the x_i with weights $w_i = \frac{1}{n}$ for $i = 1, \ldots, n$.

Comments

The definition provided at the beginning of the problem section differs slightly from the text. When n is odd, this definition is equivalent to the one provided in the text. When n is even, we pick the element following $i = \lfloor (n+1)/2 \rfloor$ to be the second median.

A change such as this to the definition of the median is needed to maked the problem well-posed. The definition of weighted median implies that the weighted median x_k is unique.

Part B

Problem Statement

Show how to compute the weighted median of n elements in $O(n \lg n)$ worst-case time using sorting.

Algorithm

The steps of our algorithm are as follows:

1. sort n elements by heapsort
2. scan from the smallest to the largest element while keeping track of the accumulative sum (acc) of the weights seen so far
   (a) if $acc \geq \frac{1}{2}$ then the current element is the weighted median
   (b) else advance to the next element
3. return the element found

Solution

Following the steps in the algorithm above, we will first analyze the correctness of the algorithm and then the running time of it. We need to demonstrate that the algorithm always returns x_k on termination.

To show that x_k is the weighted median:

$$\sum_{i<k} w_i < \frac{1}{2}(\text{part } 1) \tag{5}$$

$$\sum_{i>k} w_i < \frac{1}{2}(\text{part } 2) \tag{6}$$

Since the algorithm terminates in the k^{th} step, for part 1,

$$acc \equiv \sum_{i<k'} w_i < \frac{1}{2}Ak' < k \tag{7}$$

$$k' = k - 1 = \sum_{i<k} w_i < \frac{1}{2} \tag{8}$$

and, for part 2,

$$acc \equiv \sum_{i>k'} w_i < \frac{1}{2}Ak' < k \tag{9}$$

$$k' = k - 1 = \sum_{i>k} w_i < \frac{1}{2} \tag{10}$$

The running time of the proposed algorithm is $O(n \lg n)$. By heapsort, step 1 takes $O(n \lg n)$. Step 2 is a linear scan of n elements in the worst case, so it takes $O(n)$. Step 3 just returns the element so it takes $O(1)$ time.

$$T(n) = O(n \lg n) + O(n) + O(1) = O(n \lg n) \tag{11}$$

Therefore, the algorithm computes the weighted median of n elements in $O(n \lg n)$ worst-case time.

Part C

Problem Statement

Show how to compute the weighted median in $\theta(n)$ worst-case time using a linear-time median algorithm such as $SELECT$ from Section 10.3.

Solution

We will define an algorithm in terms of a procedure $SelWMed(A, p, r, x, y)$. The input is array A, where $1 \le p \le r \le n$, for any x and y. Its goal is to find a q and $p \le q \le r$, such that $(w_p + w_{p+1} + \ldots + w_{q-1}) \le x$ and $(w_{q+1} + w_{q+2} + \ldots + w_r) \le y$, where w is the weight of A[i].

The $SelWMed$ algorithm finds the weighted median of a sorted array in $\theta(n)$ worst-case time. SelWMed is called using the following parameters: A, p, r, x, and y.

1. Do processes 1, 2, and 3 on page 190 to find a median-of medians s with the input of A[p,r]
   $m = r$ - p + 1
   time: $T(m/5)$ and $O(m)$
2. Partition the A[p,r] at s
   time: $O(m)$
3. Let $a = (w_p + w_{p+1} + \cdots + w_{s-1}), b = (w_{s+1} + w_{s+2} + \cdots + w_r)$
   time: $O(m)$
   (a) if $a > x$ then return $SelWMed(A, p, s, x, y - b)$
   (b) elseif $b > y$ then return $SelWMed(A, s, r, x - a, y)$
   (c) else return s

   overall time: At most $T(7m/10 + 6)$

First, let us prove that $SelWMed(A, 1, n, \frac{1}{2}, \frac{1}{2})$ will return the weighted median. It is very easy to see. In $SelWMed(A, p, r, x, y)$ we can know that if we return $SelWMed(A, p, s, x, y - b)$, its goal is find t such that $p \le t \le s$ such that $(w_p + w_{p+1} + \cdots + w_{t-1}) \le x$ and $(w_{t+1} + w_{q+2} + \cdots + w_s) \le y - b$. So we have $(w_p + w_{p+1} + \cdots + w_{t-1}) \le x$ and $(w_{t+1} + w_{q+2} + \cdots + w_r) \le y$ because of $b = (w_{s+1} + w_{s+2} + \cdots + w_r)$. That means the goal of $SelWMed(A, p, s, x, y - b)$ is the same as $SelWMed(A, p, r, x, y)$. It is the same case when we return $SelWMed(A, s, r, x - a, y)$ or s. The goal of $SelWMed(A, 1, n, \frac{1}{2}, \frac{1}{2})$ is returned for $SelWMed(A, 1, n, \frac{1}{2}, \frac{1}{2})$, which is the definition of weighted median.

Second, let $T(n)$ be the worst-case time in $SelWMed(A, 1, n, \frac{1}{2}, \frac{1}{2})$. When we use s, which is provided by the processes 1, 2, and 3 on page 190, we will get two wings of array A[p,r] with no wing having length more than $7m/10 + 6$. Thus, the third step will cost no more than $T(7m/10 + 6)$.

So $T(n) \le T(n/5) + T(7n/10 + 6) + O(n)$. We can pick constant c large enough so that when $n > N$, $c(n/10 - 7)$ is larger than $O(n)$ and $T(j) \le cj$ for $j < N$. We can now see that $T(n) \le T(n/5) + T(7n/10 + 6) + c(n/10 - 7) \le cn$.

Thus, $SelWMed(A, 1, n, \frac{1}{2}, \frac{1}{2})$ can compute the weighted median in $O(n)$ worst-case time.