CS 570: Analysis of Algorithms – H7

Submitted by: Indronil Bhattacharjee

Exercise 9.2-3

Q- Write an iterative version of RANDOMIZED-SELECT.

```
ITERATIVE-RANDOMIZED-SELECT(A, p, r, i):
      while p < r do
            q = RANDOMIZED-PARTITION(A, p, r)
            k = q - p + 1
            if i=k then
                  return A[q]
            if i < k then
                  r = q - 1
            else
                  p = q
                  i = i - k
      return A[p]

RANDOMIZED-PARTITION(A, p, r):
    i = RANDOM(p, r)
    swap A[r] with A[i]
    return PARTITION(A, p, r)

PARTITION(A, p, r):
    x = A[r]
    i = p - 1
    for j = p to r - 1
        if A[j] <= x
            i = i + 1
            swap A[i] with A[j]
    swap A[i + 1] with A[r]
    return i + 1
```

Problem 9-4

Q- For n distinct elements $x_1, x_2, \ldots, x_n$ with positive weights $w_1, w_2, \ldots, w_n$ such that $\sum_{i=0}^{n} w_i = 1$, the weighted (lower) median is the element x_k satisfying

$$\sum_{xi<x} w_i < \frac{1}{2} \text{ and } \sum_{xi>x} w_i \leq \frac{1}{2}$$

(a) Argue that the median $x_1, x_2, \ldots, x_n$ is the weighted median of x_i with weights $w_i = \frac{1}{n}$ for i=1, 2, ..., n.

Let m_k be the number of x_i smaller than x_k.

When weights of $\frac{1}{n}$ are assigned to each x_i, we have $\sum_{xi<x} w_i = \frac{m_k}{n}$ and $\sum_{xi>x} w_i = \frac{n - m_k - 1}{2}$.

The only value of m_k which makes these sums $< \frac{1}{2}$ and $\leq \frac{1}{2}$ respectively is when $\lceil \frac{n}{2} \rceil - 1$, and this value of x must be the median since it has equal numbers of x_i's which are larger and smaller than it.

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

First we will use Merge Sort to sort the x_i's by value in O(n log n) time.

Let S_i be the sum of the weights of the first i elements of this sorted array and to be noted that it is O(1) time complexity to update S_i.

Compute S_1, S_2, . . . until you reach k such that $S_{k-1} < \frac{1}{2}$ and $S_k \geq \frac{1}{2}$. The weighted median is x_k.

(c) Show how to compute the weighted median in Θ(n) worst-case time using a linear-time median algorithm such as 'SELECT'.

We modify SELECT to do this in linear time.

Let x be the median of medians.

Then we have to compute $\sum_{xi < x} w_i$ and $\sum_{xi > x} w_i$ and check if either of these is larger than $\frac{1}{2}$.

If not, we should stop. If so, recurse on the collection of smaller or larger elements known to contain the weighted median. This doesn't change the runtime, so it is Θ(n).