Medians and Order Statistics

CLRS Chapter 9
What Are Order Statistics?

The \textit{k-th order statistic} is the \(k\)-th smallest element of an array.

\begin{center}
\begin{tabular}{cccccccc}
3 & 4 & 13 & 14 & 23 & 27 & 41 & 54 & 65 & 75 \\
\end{tabular}
\end{center}

8th order statistic

The \textit{lower median} is the \(\left\lfloor \frac{n}{2} \right\rfloor\)-th \textit{order statistic}

The \textit{upper median} is the \(\left\lceil \frac{n}{2} \right\rceil\)-th order statistic

If \(n\) is odd, lower and upper median are the same
What are Order Statistics?

Selecting $i^{th}$-ranked item from a collection.

- First: $i = 1$
- Last: $i = n$
- Median(s): $i = \left\lfloor \frac{n}{2} \right\rfloor, \left\lceil \frac{n}{2} \right\rceil$
Order Statistics Overview

- Assume collection is unordered, otherwise trivial. 
  \textit{find ith order stat} = A[i]

- Can sort first – $\Theta(n \lg n)$, but can do better – $\Theta(n)$.

- I can find max and min in $\Theta(n)$ time (obvious)

- Can we find any order statistic in linear time? (not obvious!)
Order Statistics Overview

How can we modify Quicksort to obtain expected-case $\Theta(n)$?

Pivot, partition, but recur only on one set of data. No join.
Using the Pivot Idea

• Randomized-Select(A[p..r],i) looking for *ith* o.s.

  if p = r
    return A[p]
  q <- Randomized-Partition(A,p,r)
  k <- q-p+1 *the size of the left partition*
  if i=k
    return A[q]
  else if i < k *then the answer is in the front*
    return Randomized-Select(A,p,q-1,i)
  else *then the answer is in the back half*
    return Randomized-Select(A,q+1,r,i-k)
Randomized Selection

- Analyzing `RandomizedSelect()`
  - Worst case: partition always 0:n-1
    \[ T(n) = T(n-1) + O(n) \]
    \[ = O(n^2) \]
    - No better than sorting!
  - “Best” case: suppose a 9:1 partition
    \[ T(n) = T(9n/10) + O(n) \]
    \[ = O(n) \quad \text{(Master Theorem, case 3)} \]
    - Better than sorting!
  - Average case: O(n) remember from quicksort
Worst-Case Linear-Time Selection

- Randomized algorithm works well in practice
- What follows is a worst-case linear time algorithm, really of theoretical interest only
- Basic idea:
  - Guarantee a good partitioning element
  - Guarantee worst-case linear time selection
- Warning: Non-obvious & unintuitive algorithm ahead!
- Blum, Floyd, Pratt, Rivest, Tarjan (1973)
Worst-Case Linear-Time Selection

- The algorithm in words:
  1. Divide \( n \) elements into groups of 5
  2. Find median of each group (How? How long?)
  3. Use Select() recursively to find median \( x \) of the \( \lceil n/5 \rceil \) medians
  4. Partition the \( n \) elements around \( x \). Let \( k = \text{rank}(x) \)
  5. \textbf{if} (\( i == k \)) \textbf{then} return \( x \)
     \textbf{if} (\( i < k \)) \textbf{then} use Select() recursively to find \( i \)th smallest element in first partition
     \textbf{else} (\( i > k \)) use Select() recursively to find \((i-k)\)th smallest element in last partition
Order Statistics: Algorithm

Select(A,n,i):
Divide input into \( \lceil n/5 \rceil \) groups of size 5.

/* Partition on median-of-medians */
medians = array of each group’s median.
pivot = Select(medians, \( \lceil n/5 \rceil \), \( \lceil n/10 \rceil \))
Left Array L and Right Array G = partition(A, pivot)

/* Find i\(^{th}\) element in L, pivot, or G */
k = |L| + 1
If i=k, return pivot
If i<k, return Select(L, k-1, i)
If i>k, return Select(G, n-k, i-k)

\[ T(n) \]
\[ O(n) \]  

All this to find a good split.

\[ T(k) \]
\[ O(1) \]
\[ T(n-k) \]  

Only one done.
Order Statistics: Analysis

\[ T(n) = T\left(\left\lfloor \frac{n}{5} \right\rfloor \right) + T\left(\max(k - 1, n - k)\right) + O(n) \]

How to simplify?
Order Statistics: Analysis

One group of 5 elements.

- Lesser Elements
- Median
- Greater Elements
Order Statistics: Analysis

All groups of 5 elements.
(And at most one smaller group.)
Order Statistics: Analysis

Definitely Lesser Elements

Definitely Greater Elements
Order Statistics: Analysis 1

Must recur on all elements outside one of these boxes.
How many?
Order Statistics: Analysis 1

\[ \left\lfloor \frac{n}{5} \right\rfloor / 2 \] full groups of 5

\[ \left\lfloor \frac{n}{5} \right\rfloor / 2 \] partial groups of 2

Count elements outside smaller box.

At most

\[ 5 \left\lfloor \frac{n}{5} \right\rfloor / 2 + 2 \left\lfloor \frac{n}{5} \right\rfloor / 2 \leq \frac{7n}{10} + 6 \]
Order Statistics: Analysis

\[ T(n) = T\left(\left\lceil \frac{n}{5} \right\rceil \right) + T\left(\frac{7n}{10} + 6 \right) + O(n) \]

A very unusual recurrence. How to solve?
Order Statistics: Analysis

Substitution: Prove \( T(n) \leq c \times n \).

\[
T(n) \leq c \times \left\lfloor \frac{n}{5} \right\rfloor + c \times \left( \frac{7n}{10} + 6 \right) + d \times n
\]

\[
\leq c \times \left( \frac{n}{5} + 1 \right) + c \times \left( \frac{7n}{10} + 6 \right) + d \times n \quad \text{Overestimate ceiling}
\]

\[
= \frac{9}{10} c \times n + 7c + d \times n
\quad \text{Algebra}
\]

\[
= c \times n - (c \times n/10 - 7c - d \times n)
\quad \text{Algebra}
\]

\[
\leq c \times n
\]

when choose \( c, d \) such that \( 0 \leq c \times n/10 - 7c - d \times n \)
Order Statistics

Why groups of 5?

Sum of two recurrence sizes must be < 1.
Grouping by 5 is smallest size that works.