TODAY

- ORDER STATISTICS - DEFINITION

- A RANDOMIZED ALGORITHM

- A DETERMINISTIC ALGORITHM
  (not randomized; "next step" of algorithm completely determined by input & previous steps.)

Problem Definition

Given: n elements $a_1, a_2, \ldots, a_n$

& index $i$

Find: $i^{th}$ smallest element.
**Related Problems**

\[ i = 1 : \text{MIN} \]

\[ i = n : \text{MAX} \]

\[ i = \left\lfloor \frac{n+1}{2} \right\rfloor : \text{MEDIAN} \]

\[ i = \left\lfloor 0.9n \right\rfloor : \text{"90\% Percentile"} \]

\[ \uparrow \]

Statistical terms.

here "order statistic"

---

**Knowledge so far:**

- Can compute MIN, MAX in \( O(n) \) time.

- But for MEDIAN? Best we know in comparison model is \( \Theta(n \log n) \).
Today: Two Linear Time Select Algs.

1. Randomized, Expected Linear
   (good workout in randomized analysis)

2. Deterministic Linear
   (good workout on divide & conquer)

Randomized Alg:

Idea: Pick random element $a$ from $A[1..n]$: 

- Partition $A[1..n]$ around $a$ (as in randomized QuickSort); 
- Now only have to recurse on one side of the partition.
So hopefully running time will be like $T(n) \leq O(n) + T(\frac{n}{2})$.

 Doesn't work out exactly like that, but close enough.
Actual Algorithm

$\textbf{Rand-select} \ (A, p, q, i)$

1. if $p = q$ return $A[p]

2. $r \gets \textbf{rand-partition} \ (A; p, q)$

   (returns $A_{\uparrow r}$)

   $A(i) \leq A(r) \quad i \leq r

   A(i) \geq A(r) \quad i \geq r

   $r$ uniform index from $[p \ldots q]$;

3. $k \gets r - p + 1

4. if $k = r$ return $A[r]

   \text{if } k < r \text{ return } \textbf{rand-select} \ (A, p, r-1, i)

   \text{if } k > r \text{ return } \textbf{rand-select} \ (A, r+1, q, i-r)$
Analysis (Similar to randomized Quicksort; but we’ll do it differently)

Let $T(n)$ = Random variable rep’ing time to find $i^{th}$ element in n elt array.

Let $X_k$ be Indicator Random variable

\[ X_k = \begin{cases} 1 & \text{if split} = (k, n-k-1) \\ 0 & \text{o.w.} \end{cases} \]

\[ \left[ \text{for } k = 0, 1, \ldots, n-1 \right] \]

Then

\[ T(n) \leq \sum_{k=0}^{n-1} X_k \cdot \max \{ T(k), T(n-k-1) \} + O(n) \]

[Concludes Step 7: Find the Recurrence]
[Begin Step 2: Analyze the recurrence]

Simplifying the recurrence, use monotonicity of $T()$.

$T(n) \leq T(n')$ if $n \leq n'$

Yields

$$\max \{ T(k), T(n-k-1) \} \leq T(n) \text{ if } 0 \leq k \leq \frac{n}{4}$$

$$\leq T \left( \frac{3n}{4} \right) \quad \frac{n}{4} \leq k \leq \frac{3n}{4}$$

$$\leq T(n) \text{ if } k \geq \frac{3n}{4}$$

$E(X_k) = \frac{1}{n}$ for every $k$.

Use these to simplify expectation of $T(n)$. 
\[ E(T(n)) \leq E \left( \sum_{k=0}^{n-1} \max\{T(k), T(n-k)\} \cdot X_k \right) + O(n) \]

\[ \leq \sum_{n=0}^{n-1} E(X_k) \cdot E(\max\{T(k), T(n-k)\}) + O(n) \]

\[ \leq \sum_{k=0}^{n-1} \frac{1}{n} E(T(n)) + \sum_{k=\frac{n}{4}}^{\frac{3n}{4}} \frac{1}{n} E(T(3n/4)) + O(n) \]

\[ + \sum_{k=\frac{3n}{4}}^{n} \frac{1}{n} \cdot E(T(n)) + O(n) \]

\[ \Rightarrow E(T(n)) \leq \frac{1}{2} E(T(n)) + \frac{1}{2} E(T(3n/4)) + O(n) \]

\[ \Rightarrow E(T(n)) \leq E(T(3n/4)) + O(n) \]

\[ \Rightarrow E(T(n)) = O(n) \text{ by Master Method.} \]
Conclusion:
- Linear Time 🧡
- Randomized 🤔
- Power of random sampling 🎉

Deterministic Algorithm

Idea: Find a "decent" partitioning element "quickly".

- "Decent" has rank between
  \[ [\cdot 3n \ldots \cdot 7n] \]
- "quickly" → linear time:
  
  or by recursion
  
  on small set (of size
  < · 3n)
Blum, Floyd, Pratt, Rivest, Tarjan's idea

\[ n \text{ elts} \]

\[ \uparrow \quad \downarrow \]

\[ \uparrow \quad \text{find median of groups} \]

\[ \text{set of medians} \]

\[ \uparrow \quad \text{find median of } B \]
Claim: This is a good pivot!!

Algorithm

\textbf{SELECT} (A, i)

* For \( j = 1 \) to \( \frac{n}{5} \)

\[ B(j) = \text{MEDIAN} \left\{ A(5j-4), A(5j-3), A(5j-2), A(5j-1), A(5j) \right\} \]

* \( x \leftarrow \text{SELECT} (B, \frac{n}{10}); \ k \leftarrow \text{rank}_A(x) \)
if i = k
    Return A(k)

i < k
    Return SELECT(A[i..k], i)

i > k
    Return SELECT(A[k+1..n], i-k)

Analysis

Claim 1: \( \frac{3n}{10} \leq k \leq \frac{7n}{10} \).

Will prove in a few minutes.

Assuming Claim:

Reformulate: \( T(n) \leq T(\frac{n}{5}) + \Theta(n) \)

\[ + \max \left\{ \max_{3n/10 \leq k \leq 7n/10} \left\lfloor T(k) \right\rfloor \right\} \]

↓ Simplified to

\( T(n) \leq T(\frac{n}{5}) + T(\frac{7n}{10}) + C \cdot n \)
By substitution will prove $T(n) \leq c' n$

**Inductive Step:**

$T(n) \leq T\left(\frac{n}{5}\right) + T\left(\frac{2n}{10}\right) + cn$

$\leq c' \left[ \frac{n}{5} + \frac{2n}{10} \right] + cn$

$= c'n - n \left[ c' \left( 1 - \frac{1}{5} - \frac{7}{10} \right) - c \right]$

$= c'n - n \left[ \frac{c'}{10} - c \right]$

$\leq c'n$ provided $c' \geq 10c$
Proof of Claim: \( \frac{3n}{10} \leq k \leq \frac{7n}{10} \)

Let's prove only this; other direction similarly.

For analysis, redraw group so that median is in middle & others' above smaller & others' below larger.

Again redraw st. median of B is in center & others on left are smaller.

Subclaim: All elements in \( \\text{region} \) are \( \leq x \).
Proof:

\[ 0 \leq 0 \leq 0 \]

\[ \# \text{ Elements} = \# \text{ columns to left} \times 3 \]

\[ = \frac{N}{10} \times 3 \]

\[ \boxed{\text{Conclusion: Simple random sampling gives excellent "pivots" (for whatever purpose!)}} \]

- Careful examination can often remove randomness ... 

- Thought for the day: When would you care?