Selection

Input: sequence $S$ of $n$ numbers, and an integer $k$ with $1 \leq k \leq n$

Output: $k$-th smallest element in $S$.

$k = 1$: smallest element in $S$
$k = n$: largest element in $S$
$k = \frac{n}{2}$: median in $S$

Algorithm:  
1. sort $S$
2. return the element at position $k$ in the sorted sequence.

Running time: $O(n \log n)$.

We will see: $O(n)$ time is possible.

:: we can find the $k$-th smallest element without sorting $S$. 
Algorithm Select $(S, k)$:

// $S$ is a sequence of numbers, $1 \leq k \leq |S|$

if $|S| = 1$ : return the only element of $S$

if $|S| > 2$:

choose an element $p$ in $S$; (pivot)

divide $S$ into $S_<$, $S_=$, $S_>$

\[
\begin{array}{ccc}
< p & = p & > p \\
S_< & S_=& & S_> \\
\end{array}
\]

if $k \leq |S_<|$ : run $\text{Select}(S_<, k)$

else if $k > |S_<| + |S_=|$:

run $\text{Select}(S_>, k - |S_<| - |S_=|)$

else return $p$
Since \( p \in S \): \( |S| \geq 1 \)

:: recursive call on a sequence of length \(< |S|\)

:: algorithm terminates

Running time: depends on the pivot \( p \).

worst case: \( S \) is sorted already

\[ k = 1 \]

in each recursive call: \( p \) = largest element

\( O(n^2) \) running time

good case: in each recursive call: \( p \) = median

this gives the recurrence \( T(n) = n + T(\frac{n}{2}) \)

\[ \therefore T(n) \leq \sum_{i=0}^{\infty} \frac{n}{2^i} = 2n = O(n). \]

how to get the "good case":

in each recursive call: choose \( p \) randomly.

intuition: on average, \( p \) will be close to the median

(for details: COMP 4804)

See my separate notes
Blum, Floyd, Pratt, Rivest, Tarjan (1973): selection in \( O(n) \) worst-case time.

**General approach:**

- **Input:** sequence \( S \) of \( n \) numbers, \( 1 \leq k \leq n \).
  - (assume all numbers distinct)

**Assume:** in \( O(n) \) time, we can find an element \( p \) in \( S \), such that \( |S_p| \leq \alpha n \) and \( |S_p| \leq \alpha n \), where \( 0 < \alpha < 1 \) is a constant.

<table>
<thead>
<tr>
<th>&lt;p</th>
<th>p</th>
<th>&gt;p</th>
</tr>
</thead>
</table>

\[ \leq \alpha n \] \[ \leq \alpha n \]

Then the running time \( T(n) \) satisfies

\[ T(n) = n + T(\alpha n) \]

\[ \therefore T(n) = n \left(1 + \alpha + \alpha^2 + \alpha^3 + \ldots\right) \]

\[ \leq \frac{1}{1-\alpha} n = O(n) \]
Here is the algorithm:

**Step 1:** Divide the input sequence into \( \frac{n}{5} \) groups, each of length 5.

**Step 2:** for \( i = 1, 2, \ldots, \frac{n}{5} \) : compute the median of the \( i \)-th group; call this median \( m_i \)

**Step 3:** Compute the median \( p \) of \( m_1, m_2, \ldots, m_{n/5} \).

**Step 4:** Use \( p \) as the pivot; proceed as on page 34.

Why is \( p \) a good pivot:

How many of the \( n \) elements are \( \geq p \)?

In the following figure, each column is a group of 5 elements; the middle row is the sequence of medians; for the purpose of the figure, assume each column is sorted, and the middle row is sorted from left to right from bottom to top.
all elements in the $3 \times \frac{n}{10}$ rectangle are $\geq p$

$\therefore \geq \frac{3}{10} n$ elements are $\geq p$

$\therefore \leq \frac{7}{10} n$ elements are $< p$

$\therefore |S_\leq| \leq \frac{7}{10} n$

By a symmetric argument: $|S_\geq| \leq \frac{7}{10} n$

$\therefore$ on page 36, we can take $\alpha = \frac{7}{10}$
Define $T(n) = \text{worst-case running time on an input of length } n.$

$$T(n) = O(n) \quad \leftarrow \text{Step 1}$$

$$+ O(n) \quad \leftarrow \text{Step 2}$$

$$+ ? \quad \leftarrow \text{Step 3}$$

$$+ O(n) + T(\frac{7}{10}n) \quad \leftarrow \text{Step 4}$$

How to do Step 3: Recursively compute the $\frac{n}{10}$-th smallest element of the sequence $m_1, \ldots, m_{n/5};$ this takes $T(\frac{n}{5})$ time.

We obtain the recurrence

$$T(n) = n + T(\frac{n}{5}) + T(\frac{7}{10}n).$$

How to solve this:

- unfolding: becomes very messy,
- Master Theorem: does not cover this recurrence,
- use induction to show that $T(n) = O(n).$
Claim: $T(n) \leq cn$ for some constant $c$.

Proof: By choosing $c$ sufficiently large, the claim is true for "small" $n$. (This is the base case of the induction.)

Let $n$ be "large" and assume $T(m) \leq cm$ for all $1 \leq m < n$.

Then $T(n) = n + T\left(\frac{n}{5}\right) + T\left(\frac{7}{10} n\right)$

\[ \leq n + c \cdot \frac{n}{5} + c \cdot \frac{7}{10} n \]

\[ = n + \frac{9}{10} cn \]

\[ \leq cn \text{ if } n \leq \frac{1}{10} cn \text{ iff } c \geq 10. \]

Conclusion: The $k$-th smallest element in a sequence of $n$ numbers can be computed in $O(n)$ time.