Selection

Select(A,i,n) – Given a set of \( n \) numbers A, find the \( i \) th smallest.

Example

3 18 1 8 4 47 45 10 23

- \( i = 1 \) – 1 minimum
- \( i = 9 \) – 47 maximum
- \( i = 5 \) – 10 median
- \( i = 7 \) – 23

What is a straightforward algorithm for Selection?
Selection

Select(A,i,n) – Given a set of \( n \) numbers A, find the \( i \) th smallest.

Example

3 18 1 8 4 47 45 10 23

- \( i = 1 \) – 1 minimum
- \( i = 9 \) – 47 maximum
- \( i = 5 \) – 10 median
- \( i = 7 \) – 23

What is a straightforward algorithm for Selection?

- Sort the numbers and return \( A[i] \).
- \( O(n \log n) \) time.
Selection

Question: Is selection (median) harder than sorting? Is it necessary to sort to find the median?
Selection

Question: Is selection (median) harder than sorting? Is it necessary to sort to find the median?

Answer: No. We can find the median in $O(n)$ time.
A recursive strategy

- Pick an element to be the pivot $p$.
- Split the items into 2 sets
  - $L$ – the set of items less than or equal to the pivot
  - $H$ – the set of items greater than the pivot.
- Recurse on the appropriate set.
**Deterministic Selection**

\[
\text{Select}(A,i,n)\\
1 \quad \textbf{if} \quad (n = 1)\\
2 \quad \textbf{then return} \quad A[1]\\
3 \quad p = \text{median}(A)\\
4 \quad 5\\
6 \quad L = \{x \in A : x \leq p\}\\
7 \quad H = \{x \in A : x > p\}\\
8 \quad \textbf{if} \quad i \leq |L|\\
9 \quad \textbf{then Select}(L, i, |L|)\\
10 \quad \textbf{else Select}(H, i - |L|, |H|)\
\]
Analysis

• Let $M(n)$ be the time to find the median
• Let $T(n)$ be the time for selection
• Splitting the items into $L$ and $H$ takes $O(n)$ time.

\[ T(n) = T(n/2) + M(n) + O(n). \]

• If $M(n) = O(n)$, then $T(n) = O(n)$
• But why should $M(n) = O(n)$, and isn’t the reasoning circular?
**Modified Goal**

**New Goal:** Suppose that, in $O(n)$ time, we find a number “close” to the median, then what happens.

- Question 1: What if, in $O(n)$ time, we found a number in the middle half?
- Question 2: What if we used our selection algorithm recursively to help find a number in the middle half?
Modified Goal

New Goal: Suppose that, in $O(n)$ time, we find a number “close” to the median, then what happens.

• Question 1: What if, in $O(n)$ time, we found a number in the middle half?

• Question 2: What if we used our selection algorithm recursively to help find a number in the middle half?

Answer 1: If we found a number in the middle half, we would have the recurrence

$$T(n) \leq T(3n/4) + O(n)$$

which by the master method solves to $O(n)$. 
Deterministic Selection (2)

```plaintext
SELECT(A,i,n)

1 if (n = 1)
2 then return A

3 Split the items into $\lfloor n/5 \rfloor$ groups 5 (and one more group).
Call these groups $G_1, G_2, \ldots, G_{\lfloor n/5 \rfloor}$

4 Find the median $m_i$ of each $G_i$

5 Recursively compute the median of medians,
$p = SELECT(\{m_1, \ldots, m_{\lfloor n/5 \rfloor}\}, \lfloor n/10 \rfloor, \lfloor n/5 \rfloor)$

6 $L = \{x \in A: x \leq p\}$
$H = \{x \in A: x > p\}$

7 if $i \leq |L|$
8 then $SELECT(L, i, |L|)$
9 else $SELECT(H, i - |L|, |H|)$
```
Proof

Correctness  Need to prove that lines 3-6 return an item in the middle half.

Running time

\[ T(n) \leq T(3n/4) + T(n/5) + O(n) \]