Algorithm: Sel(numbers,k)

Given: numbers $x_1, ..., x_n$, k, output k’th smallest number.
Algorithm: Sel(numbers,k)

Given: numbers $x_1, ..., x_n$, k, output k’th smallest number.

Sort the numbers! $O(n \log n)$ time.

Can we do better?
Given: $x_1, ..., x_n$, $k$, output $k$'th smallest number.

Algorithm: Sel(numbers, k)

1. Partition numbers into sets of size 3
Algorithm: Sel(numbers, k)

1. Partition numbers into sets of size 3
2. Sort each set
1. Partition numbers into sets of size 3
2. Sort each set
3. $w = \text{Sel}(\quad , n/6)$
1. Partition numbers into sets of size 3
2. Sort each set
3. \( w = \text{Sel}(\_\_\_, n/6) \)

\[
egin{align*}
S_L(w) &= \{ x_i \mid x_i < w \} \\
S_E(w) &= \{ x_i \mid x_i = w \} \\
S_G(w) &= \{ x_i \mid x_i > w \}
\end{align*}
\]

Can be computed in linear time
1. Partition numbers into sets of size 3
2. Sort each set
3. \( w = \text{Sel}(\ ) , n/6) \)

\[
\begin{align*}
S_L(w) &= \{ x_i \mid x_i < w \} \\
S_E(w) &= \{ x_i \mid x_i = w \} \\
S_G(w) &= \{ x_i \mid x_i > w \}
\end{align*}
\]

- Can be computed in linear time

4. if \( k \leq |S_L(w)| \), output \( \text{Sel}(S_L(w), k) \)
else if \( k \leq |S_L(w)| + |S_E(w)| \), output \( w \)
else output \( \text{Sel}(S_G(w), k - |S_L(w)|-|S_E(w)|) \)
Algorithm: Sel(numbers, k)

\[ w = \text{Sel}(), \frac{n}{6} \]

If \( k \leq |S_L(w)| \), output \( \text{Sel}(S_L(w), k) \)

Else if \( k \leq |S_L(w)| + |S_E(w)| \), output \( w \)

Else output \( \text{Sel}(S_G(w), k - |S_L(w)| - |S_G(w)|) \)
Algorithm: Sel(numbers, k)

\[ w = \text{Sel}(\text{numbers}, \frac{n}{6}) \]

if \( k \leq |S_L(w)| \), output \( \text{Sel}(S_L(w), k) \)
else if \( k \leq |S_L(w)| + |S_E(w)| \), output \( w \)
else output \( \text{Sel}(S_G(w), k - |S_L(w)| - |S_G(w)|) \)
Algorithm: Sel(numbers, k)

1. Partition numbers into sets of size 3
2. Sort each set
3. \( w = \text{median of } \)
   \[ S_L(w) = \{ x_i \mid x_i < w \} \]
   \[ S_E(w) = \{ x_i \mid x_i = w \} \]
   \[ S_G(w) = \{ x_i \mid x_i > w \} \]

4. if \( k \leq |S_L(w)| \), output Sel(S_L(w), k)
   else if \( k \leq |S_L(w)| + |S_E(w)| \), output w
   else output Sel(S_G(w), k - |S_L(w)| - |S_E(w)|)

\[ T(n) = T(n/3) + T(2n/3) + O(n) \]

Can be computed in linear time

\[ |S_L(w)| + |S_E(w)|, |S_G(w)| + |S_E(w)|, \text{ at least } 2(n/6) = n/3 \]
Algorithm: Sel(numbers, k)

1. Partition numbers into sets of size 3
2. Sort each set (only in our heads)
3. \( w = \text{median of } numbers \)
\
\[
\begin{align*}
|S_L(w)| + |S_E(w)|, |S_G(w)| + |S_E(w)|, \text{ at least } 2(n/6) = n/3
\end{align*}
\]

4. if \( k \leq |S_L(w)| \), output Sel(S_L(w), k)
else if \( k \leq |S_L(w)| + |S_E(w)| \), output \( w \)
else output Sel(S_G(w), k - |S_L(w)| - |S_E(w)|)

1. Partition numbers into sets of size 3
2. Sort each set (only in our heads)
3. \( w = \text{median of } numbers \)

\[
\begin{align*}
S_L(w) = \{ x_i \mid x_i < w \} \\
S_E(w) = \{ x_i \mid x_i = w \} \\
S_G(w) = \{ x_i \mid x_i > w \}
\end{align*}
\]

\[ T(n) = T(n/3) + T(2n/3) + O(n) \]
so
\[ T(n) = O(n \log n) \]

(what's the point???)
Recurrences

\[ T(n) = T(\gamma n) + T(\beta n) + n \]
Algorithm: Sel(numbers, k)

1. Partition numbers into sets of size 5
2. Sort each set
3. \( w = \text{median of} \)
   \[
   \begin{align*}
   S_L(w) &= \{x_i \mid x_i < w\} \\
   S_E(w) &= \{x_i \mid x_i = w\} \\
   S_G(w) &= \{x_i \mid x_i > w\}
   \end{align*}
   \] Can be computed in linear time
4. if \( k \leq |S_L(w)| \), output \( \text{Sel}(S_L(w), k) \)
   else if \( k \leq |S_L(w)| + |S_E(w)| \), output \( w \)
   else output \( \text{Sel}(S_G(w), k - |S_L(w)| - |S_E(w)|) \)
Algorithm: Sel(numbers, k)

1. Partition numbers into sets of size 5
2. Sort each set
3. \( w = \text{median of } S_L(w) \) if \( |S_L(w)| + |S_E(w)| \) is at least \( n/5 \)

\[ S_L(w) = \{x_i \mid x_i < w\} \]
\[ S_E(w) = \{x_i \mid x_i = w\} \]
\[ S_G(w) = \{x_i \mid x_i > w\} \]

4. if \( k \leq |S_L(w)| \), output \( \text{Sel}(S_L(w), k) \)
   else if \( k \leq |S_L(w)| + |S_E(w)| \), output \( w \)
   else output \( \text{Sel}(S_G(w), k - |S_L(w)| - |S_E(w)|) \)

Can be computed in linear time
Algorithm: Sel(numbers, k)

1. Partition numbers into sets of size 5
2. Sort each set
3. \( w = \text{median of} \)
   \[ S_L(w) = \{ x_i \mid x_i < w \}, \quad S_E(w) = \{ x_i \mid x_i = w \}, \quad S_G(w) = \{ x_i \mid x_i > w \} \] Can be computed in linear time
4. if \( k \leq |S_L(w)| \), output Sel(S_L(w), k)
   else if \( k \leq |S_L(w)| + |S_E(w)| \), output w
   else output Sel(S_G(w), k - |S_L(w)| - |S_E(w)|)

\[ |S_L(w)| + |S_E(w)|, \quad |S_G(w)| + |S_E(w)|, \quad \text{at least } 3n/10 \]
1. Partition numbers into sets of size 5
2. Sort each set
3. \( w = \text{median of } \)
4. if \( k \leq |S_L(w)| \), output \( \text{Sel}(S_L(w), k) \)
   else if \( k \leq |S_L(w)| + |S_E(w)| \), output \( w \)
   else output \( \text{Sel}(S_G(w), k - |S_L(w)| - |S_E(w)|) \)