Linear-time Selection
Linear-time Selection

- Given an array $A$ of $n$ elements and an integer $1 \leq k \leq n$, find the $k^{th}$ smallest element in $A$
- Naïve algorithm, sort $A$ and pick the $k^{th}$ element in the sorted array $\Rightarrow \Theta(n \log n)$
- Select and remove the smallest element $k$ times $\Rightarrow \Theta(nk)$
- Another quick-sort-like algorithm
  - Pick the first element (pivot)
  - Place it in its position in the array
  - Recursively process one subarray
Quick-sort-like Algorithm

$T(n) = \Theta(n^2)$

How to choose a good pivot?
Median of Fives

A = \{93, 78, 32, 81, 20, 92, 49, 28, 62, 34, 41, 51, 83, 56, 67, 90, 36, 31, 46, 50, 43, 88, 59, 1, 7\}
Median of Fives

A = \{93, 78, 32, 81, 20, 92, 49, 28, 62, 34, 41, 51, 83, 56, 67, 90, 36, 31, 46, 50, 43, 88, 59, 1, 7\}

1. Partition into groups of 5

<p>| | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>93</td>
<td>92</td>
<td>41</td>
<td>90</td>
<td>43</td>
</tr>
<tr>
<td>78</td>
<td>49</td>
<td>51</td>
<td>36</td>
<td>88</td>
</tr>
<tr>
<td>32</td>
<td>28</td>
<td>83</td>
<td>31</td>
<td>59</td>
</tr>
<tr>
<td>81</td>
<td>62</td>
<td>56</td>
<td>46</td>
<td>1</td>
</tr>
<tr>
<td>20</td>
<td>34</td>
<td>67</td>
<td>50</td>
<td>7</td>
</tr>
</tbody>
</table>
Median of Fives

A = {93, 78, 32, 81, 20, 92, 49, 28, 62, 34, 41, 51, 83, 56, 67, 90, 36, 31, 46, 50, 43, 88, 59, 1, 7}

2. Sort each sublist
Median of Fives

A = \{93, 78, 32, 81, 20, 92, 49, 28, 62, 34, 41, 51, 83, 56, 67, 90, 36, 31, 46, 50, 43, 88, 59, 1, 7\}

3. Find the median of each sublist
Median of Fives

A = {93, 78, 32, 81, 20, 92, 49, 28, 62, 34, 41, 51, 83, 56, 67, 90, 36, 31, 46, 50, 43, 88, 59, 1, 7}

4. Recursively find the median of the medians

A` = {78, 49, 56, 45, 43}

Median of medians (m*) = 49

Partition A around m* and recursively process one side
Algorithm Pseudo-code

1. SELECT($A, n, k$)
2. if ($n \leq 5$) then sort $A$ and return $k^{th}$ element
3. Partition $A$ into groups of 5
4. $M \leftarrow$ Find the median of each group
5. $m^* = \text{SELECT}(M, \frac{n}{5}, \frac{n}{10})$
6. Partition $A$ around $m^*$, let it be at position $i$
7. if ($i = k$) then return $m^*$
8. if ($i > k$) then return $\text{SELECT}(A[1, i - 1], i - 1, k)$
9. if ($i < k$) then return $\text{SELECT}(A[i + 1, n], n - i, k - i)$
Size of Sublist

A

\[ m^* \]
*Size of Sublist*

- \( \left\lfloor \frac{n}{5} \right\rfloor \) groups each of size 5
- \( m^* \) is larger than half of them
- \( m^* \) is larger than \( \frac{m}{10} \) groups
- \( m^* \) is larger than at least 3 elements in each group
- Size of each of the two sublists is \( \left\lfloor \frac{3}{10} n, \frac{7}{10} n \right\rfloor \)
- Worst-case scenario, we prune \( \frac{3n}{10} \) elements and recursively process \( \frac{7n}{10} \)
Recurrence Relation

\[ T(n) = \begin{cases} 
    a & ; n \leq 5 \\
    T \left( \frac{n}{5} \right) + T \left( \frac{7n}{10} \right) + an & ; n > 5
\end{cases} \]

- Can we apply the Master theorem?
- Let’s try recursive tree expansion
Recurrence Tree

\[ T(n) \]
Recurrence Tree

\[
\begin{align*}
\text{an} & \\
T\left(\frac{n}{5}\right) & \quad T\left(\frac{7n}{10}\right)
\end{align*}
\]
Recurrence Tree

\[ a_n \]

\[ \frac{a_n}{5} \] \hspace{1cm} \frac{7a_n}{10} \]

\[ T \left( \frac{n}{25} \right) \quad T \left( \frac{7n}{50} \right) \]

\[ T \left( \frac{7n}{50} \right) \quad T \left( \frac{49n}{100} \right) \]
Recurrence Tree

\[
\begin{align*}
&a_n \\
&\frac{a_n}{5} \\
&T\left(\frac{n}{25}\right) \quad T\left(\frac{7n}{50}\right) \\
&\frac{7a_n}{10} \\
&T\left(\frac{7n}{50}\right) \quad T\left(\frac{49n}{100}\right) \\
&\frac{9n}{10} \\
&\frac{81n}{100}
\end{align*}
\]
Running Time

$T(n) = \sum_{i=1}^{??} \left( \frac{9}{10} \right)^i n = n \sum_{i=1}^{??} \left( \frac{9}{10} \right)^i$

$T(n) \leq n \left( \frac{1}{1 - \frac{9}{10}} \right) \leq 10n$

$T(n) = \Theta(n)$
Proof by Induction

- We want to prove that $T(n) = O(n)$
- $T(n) \leq cn$, for $c > 0$ and $n \geq n_0$
- Base case: $T(n) = a$ for $n \leq 5$
- Setting $c \geq a$ satisfies the base case
- Assume that $T(n) \leq cn$ is true for all $n < m$
- We want to prove that it is true for $n = m + 1$
- $T(m + 1) = T\left(\frac{m+1}{5}\right) + T\left(\frac{7(m+1)}{10}\right) + a(m + 1)$
- $T(m + 1) \leq c\left(\frac{m+1}{5}\right) + c\left(\frac{7(m+1)}{10}\right) + a(m + 1)$
Proof by Induction

We want to prove that
\[ c \left( \frac{m+1}{5} \right) + c \left( \frac{7(m+1)}{10} \right) + a(m+1) \leq c(m+1) \]

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

\[ \frac{c}{10} \geq a \]

\[ c \geq 10a \]

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

- Express problems as divide and conquer algorithms
- Analyze the running time of divide and conquer algorithms
- Optimize and improve the worst-case running time of divide and conquer algorithms
- Reading: Chapter 4, Section 9.3