Median and Order Statistics
Order Statistics (Selection Problem)

• Select the $i$-th smallest of $n$ elements
  (select the element with rank $i$)
  – $i = 1$: minimum
  – $i = n$: maximum
  – $i = \lceil (n+1)/2 \rceil$ or $\lfloor (n+1)/2 \rfloor$: median

• Naive algorithm: Sort and index $i$-th element

\[
T(n) = \Theta(n \log n) + \Theta(1)
\]

\[
= \Theta(n \log n)
\]

using merge sort or heapsort (not quicksort)
Selection in Expected Linear Time

- Randomized algorithm
- Divide and conquer
- Similar to randomized quicksort
  - Like quicksort: Partitions input array recursively
  - Unlike quicksort:
    - Only works on one side of the partition
    - Quicksort works on both sides of the partition
- Expected running times:
  - SELECT: \( E[n] = \Theta(n) \)
  - QUICKSORT: \( E[n] = \Theta(n \log n) \)
Selection in Expected Linear Time (example)

Select the $i = 7$th smallest

Partition:

select the $7-3=4$th smallest element recursively
Selection in Expected Linear Time

R-SELECT(A, p, r, i)

if \( p = r \) then
  return \( A[p] \)

\( q \leftarrow \text{R-PARTITION}(A, p, r) \)

\( k \leftarrow q - p + 1 \)

if \( i \leq k \) then
  return R-SELECT(A, p, q, i)

else
  return R-SELECT(A, q+1, r, i-k)

\( x = \text{pivot} \)

\( \leq x \) (k smallest elements) \( \geq x \)

p \hspace{1cm} q \hspace{1cm} r
Selection in Expected Linear Time

- All elements in $L \leq$ all elements in $R$
- $L$ contains $|L| = q-p+1 = k$ smallest elements of $A[p...r]$
  
  if $i \leq |L| = k$ then
  
  search $L$ recursively for its $i$-th smallest element

  else
  
  search $R$ recursively for its $(i-k)$-th smallest element
Selection in Expected Linear Time

• Excellent algorithm in practise
• Worst-case: $T(n) = T(n-1) + \Theta(n) \Rightarrow T(n) = \Theta(n^2)$
  – Worse than sorting
  – e.g., occurs when
    – $i = 1$ and
      – Partition returns $q = r - 1$ at each level of recursion
• Best-case: $T(n) = T(n/2) + \Theta(n) \Rightarrow T(n) = \Theta(n)$
Average-Case Analysis of Randomized Select

Recall: \( P(|L|=i) = \begin{cases} 
2/n & \text{for } i = 1 \\
1/n & \text{for } i = 2, 3, \ldots, n-1 
\end{cases} \)

Upper bound: Assume \( i \)-th element always falls into the larger part

\[
T(n) \leq \frac{1}{n} T(\max(1, n-1)) + \frac{1}{n} \sum_{q=1}^{n-1} T(\max(q, n-q)) + O(n)
\]

But, \( \frac{1}{n} T(\max(1, n-1)) = \frac{1}{n} T(n-1) = \frac{1}{n} O(n^2) = O(n) \)

\[
\therefore T(n) \leq \frac{1}{n} \sum_{q=1}^{n-1} T(\max(q, n-q)) + O(n)
\]
Average-Case Analysis of Randomized Select

\[ T(n) \leq \frac{1}{n} \sum_{q=1}^{n-1} T(\max(q, n-q)) + O(n) \]

\[ \max(q, n-q) = \begin{cases} 
q & \text{if } q \geq \lfloor n/2 \rfloor \\
q & \text{if } q < \lfloor n/2 \rfloor 
\end{cases} \]

- \( n \) is odd: \( T(k) \) appears twice for \( k = \lfloor n/2 \rfloor + 1, \lfloor n/2 \rfloor + 2, \ldots, n-1 \)
- \( n \) is even: \( T(\lfloor n/2 \rfloor) \) appears once \( T(k) \) appears twice for \( k = \lfloor n/2 \rfloor + 1, \lfloor n/2 \rfloor + 2, \ldots, n-1 \)

Hence, in both cases:

\[ \sum_{q=1}^{n-1} T(\max(q, n-q)) + O(n) \leq \sum_{q=\lfloor n/2 \rfloor}^{n-1} T(q) \]

\[ T(n) \leq \frac{2}{n} \sum_{q=\lfloor n/2 \rfloor}^{n-1} T(q) + O(n) \]
Average-Case Analysis of Randomized Select

\[ T(n) \leq \frac{2}{n} \sum_{q=\lceil n/2 \rceil}^{n-1} T(q) + O(n) \]

By substitution guess \( T(n) = O(n) \)

Inductive hypothesis: \( T(k) \leq ck, \quad \forall \ k < n \)

\[ T(n) \leq 2 \sum_{k=\lceil n/2 \rceil}^{n-1} ck + O(n) \]

\[ = \frac{2c}{n} \left( \sum_{k=1}^{n-1} k - \sum_{k=1}^{\lceil n/2 \rceil - 1} k \right) + O(n) \]

\[ = \frac{2c}{n} \left( \frac{1}{2} n (n-1) - \frac{1}{2} \left( \frac{n}{2} \right) \left( \frac{n}{2} - 1 \right) \right) + O(n) \]
Average-Case Analysis of Randomized Select

\[ T(n) \leq \frac{2c}{n} \left[ \frac{1}{2} n(n-1) - \frac{1}{2} \left\lfloor \frac{n}{2} \right\rfloor \left\lfloor \frac{n}{2} - 1 \right\rfloor \right] + O(n) \]

\[ \leq c(n-1) - \frac{c}{4} n + \frac{c}{2} + O(n) \]

\[ = cn - \frac{c}{4} n - \frac{c}{2} + O(n) \]

\[ = cn - \left( \frac{c}{4} n + \frac{c}{2} \right) - O(n) \]

\[ \leq cn \]

\textit{since we can choose c large enough so that } (\frac{cn}{4}+c/2) \textit{ dominates } O(n) \]
Summary of Randomized Order-Statistic Selection

• Works fast: linear expected time
• Excellent algorithm in practise
• But, the worst case is very bad: $\Theta(n^2)$

Q: Is there an algorithm that runs in linear time in the worst case?

A: Yes, due to Blum, Floyd, Pratt, Rivest & Tarjan[1973]

Idea: Generate a good pivot recursively..
Selection in Worst Case Linear Time

\[ \text{SELECT}(S, n, i) \quad \triangleright \text{return } i\text{-th element in set } S \text{ with } n \text{ elements} \]

if \( n \leq 5 \) then

\[ \text{SORT } S \text{ and return the } i\text{-th element} \]

DIVIDE \( S \) into \( \lceil n/5 \rceil \) groups

\( \triangleright \) first \( \lceil n/5 \rceil \) groups are of size 5, last group is of size \( n \mod 5 \)

FIND median set \( M=\{m_1, \ldots, m_{\lceil n/5 \rceil}\} \quad \triangleright m_j: \text{median of } j\text{-th group} \)

\( x \leftarrow \text{SELECT}(M, n/5, \lceil (\lceil n/5 \rceil +1)/2 \rceil) \)

PARTITION set \( S \) around the pivot \( x \) into \( L \) and \( R \)

if \( i \leq |L| \) then

return \( \text{SELECT}(L, |L|, i) \)

else

return \( \text{SELECT}(R, n-|L|, i-|L|) \)
Choosing the Pivot

1. Divide $S$ into groups of size 5
Choosing the Pivot

1. Divide $S$ into groups of size 5
2. Find the median of each group
Choosing the Pivot

1. Divide $S$ into groups of size 5
2. Find the median of each group
3. Recursively select the median $x$ of the medians
Choosing the Pivot

At least half of the medians $\geq x$

Thus $m = \left\lceil \frac{n}{5} \right\rceil / 2$ groups contribute 3 elements to $R$ except possibly the last group and the group that contains $x$

$|R| \geq 3 \left( m - 2 \right) \geq \frac{3n}{10} - 6$
Similarly |

\[ |L| \geq \frac{3n}{10} - 6 \]

Therefore, SELECT is recursively called on at most

\[ n - \left( \frac{3n}{10} - 6 \right) = \frac{7n}{10} + 6 \text{ elements} \]
Selection in Worst Case Linear Time

\[ \text{SELECT}(S, n, i) \quad \triangleright \text{return } i\text{-th element in set } S \text{ with } n \text{ elements} \]

if \( n \leq 5 \) then

\[ \text{SORT } S \text{ and return the } i\text{-th element} \]

\[ \Theta(n) \left\{ \begin{array}{l}
\text{DIVIDE } S \text{ into } \lceil n/5 \rceil \text{ groups} \\
\text{first } \lfloor n/5 \rfloor \text{ groups are of size 5, last group is of size } n \mod 5 \\
\text{FIND median set } M=\{m_1, \ldots, m_{\lfloor n/5 \rfloor}\} \triangleright m_j: \text{median of } j\text{-th group} \\
\end{array} \right. \]

\[ T\left(\left\lfloor n/5 \right\rfloor\right) \{ x \leftarrow \text{SELECT}(M, \left\lfloor n/5 \right\rfloor, (\left\lfloor n/5 \right\rfloor+1)/2) \}
\]

\[ \Theta(n) \left\{ \begin{array}{l}
\text{PARTITION set } S \text{ around the pivot } x \text{ into } L \text{ and } R \\
\end{array} \right. \]

\[ T\left(\frac{7n}{10} + 6\right) \left\{ \begin{array}{l}
\text{if } i \leq |L| \text{ then} \\
\quad \text{return } \text{SELECT}(L, |L|, i) \\
\text{else} \\
\quad \text{return } \text{SELECT}(R, n-|L|, i-|L|) \\
\end{array} \right. \]
Selection in Worst Case Linear Time

Thus recurrence becomes

\[ T(n) \leq T \left( \left\lceil \frac{n}{5} \right\rceil \right) + T \left( \frac{7n}{10} + 6 \right) + \Theta(n) \]

Guess \( T(n) = O(n) \) and prove by induction

Inductive step: \( T(n) \leq c \left\lceil \frac{n}{5} \right\rceil + c \left( \frac{7n}{10} + 6 \right) + \Theta(n) \)
\leq cn/5 + c + 7cn/10 + 6c + \Theta(n)
= 9cn/10 + 7c + \Theta(n)
= cn – [c(n/10 – 7) – \Theta(n)] \leq cn \text{ for large } c

Work at each level of recursion is a constant factor (9/10) smaller