Linear Time Selection

These lecture slides are adapted from CLRS 10.3.

Where to Build a Road?

Given $(x,y)$ coordinates of $N$ houses, where should you build a road parallel to the x-axis to minimize the construction cost of building driveways?

- $n_1 =$ nodes on top.
- $n_2 =$ nodes on bottom.

Decreases total cost by $(n_2 - n_1)$ $\epsilon$

Where to Build a Road?

Given $(x,y)$ coordinates of $N$ houses, where should you build a road parallel to the x-axis to minimize the construction cost of building driveways?

- $n_1 =$ nodes on top.
- $n_2 =$ nodes on bottom.
Where to Build a Road?

Given \((x,y)\) coordinates of \(N\) houses, where should you build road parallel to \(x\)-axis to minimize construction cost of building driveways?

- \(n_1\) = nodes on top.
- \(n_2\) = nodes on bottom.

Solution: put street at median of \(y\) coordinates.

Order Statistics

Given \(N\) linearly ordered elements, find \(i^{th}\) smallest element.

- **Min:** \(i = 1\)
- **Max:** \(i = N\)
- **Median:** \(i = \lceil (N+1)/2 \rceil \) and \(\lfloor (N+1)/2 \rfloor \)
- **O**(\(N\)) for min or max.
- **O**(\(N \log N\)) comparisons by sorting.
- **O**(\(N \log i\)) with heaps.

Can we do in worst-case \(O(N)\) comparisons?

- Surprisingly, yes. (Blum, Floyd, Pratt, Rivest, Tarjan, 1973)

Assumption to make presentation cleaner.

- All items have distinct values.

Select

Similar to quicksort, but throw away useless "half" at each iteration.

- Select \(i^{th}\) smallest element from \(a_1, a_2, \ldots, a_N\).

\[
\text{Select}(i^{th}, N, a_1, a_2, \ldots, a_N) = \begin{cases} 
 x & \text{if } (i == k) \\
 \text{Select}(i^{th}, k-1, b_1, b_2, \ldots, b_{k-1}) & \text{if } (i < k) \\
 \text{Select}((i-k)^{th}, N-k, c_1, c_2, \ldots, c_{N-k}) & \text{if } (i > k)
\end{cases}
\]

\(x\) = partition element

Want to choose \(x\) so that \(x\) is (roughly) the \(i^{th}\) largest.
Partition

Partition().
- Divide N elements into $\lfloor N/5 \rfloor$ groups of 5 elements each, plus extra.
- Brute force sort each of the 5-element groups.

Find $x =$ "median of medians" by Select() on $\lfloor N/5 \rfloor$ medians.

Select

Select ($i^{th}$, N, a₁, a₂, ..., aₙ)

if (N is small) use mergesort

Divide a[] into groups of 5, and let $m_1$, $m_2$, ..., $m_{N/5}$ be list of medians.

$x \leftarrow \text{Select}(N/10, m_1, m_2, ..., m_{N/5})$

$k \leftarrow \text{rank}(x)$

if (i == k) // Case 1
    return $x$

else if (i < k) // Case 2
    $b[] \leftarrow$ all items of a[] less than $x$
    return Select($i^{th}$, k-1, $b_1$, $b_2$, ..., $b_{k-1}$)

else if (i > k) // Case 3
    $c[] \leftarrow$ all items of a[] greater than $x$
    return Select(($i-k)^{th}$, N-k, $c_1$, $c_2$, ..., $c_{N-k}$)
Selection Analysis

**Selection Analysis**

**Crux of proof:** delete roughly 30% of elements by partitioning.

- At least 1/2 of 5 element medians ≤ x
  - at least \(\lceil N / 5 \rceil / 2 = \lceil N / 10 \rceil\) medians ≤ x
- At least 3 \(\lfloor N / 10 \rfloor\) elements ≤ x.

```latex
\begin{align*}
\text{median of medians} & \quad 14 \quad 09 \quad 05 \quad 03 \quad 02 \quad 12 \quad 01 \quad 17 \quad 20 \quad 04 \quad 36 \\
& \quad 22 \quad 10 \quad 06 \quad 11 \quad 25 \quad 16 \quad 13 \quad 24 \quad 31 \quad 07 \quad 27 \\
& \quad 28 \quad 23 \quad 38 \quad 15 \quad 40 \quad 19 \quad 18 \quad 43 \quad 32 \quad 35 \quad 08 \\
& \quad 29 \quad 39 \quad 50 \quad 26 \quad 53 \quad 30 \quad 41 \quad 46 \quad 33 \quad 49 \quad 21 \\
& \quad 45 \quad 44 \quad 52 \quad 37 \quad 54 \quad 53 \quad 48 \quad 47 \quad 34 \quad 51
\end{align*}
```

Now, solve recurrence.

- Apply master theorem?
- Assume \(N\) is a power of 2?
- Assume \(C(N)\) is monotone non-decreasing?
Selection Analysis

Analysis of selection recurrence.
- T(N) = # comparisons on a file of size $\leq N$.
- T(N) is monotone, but C(N) is not!

\[
T(N) = \begin{cases} 
20cN & \text{if } N < 50 \\
T(\lceil N/5 \rceil) + T(N - 3\lceil N/10 \rceil) + cN & \text{otherwise}
\end{cases}
\]

Claim: $T(N) \leq 20cN$.
- Base case: $N < 50$.
- Inductive hypothesis: assume true for 1, 2, \ldots, N-1.
- Induction step: for $N \geq 50$, we have:

\[
T(N) \leq T(\lceil N/5 \rceil) + T(N - 3\lceil N/10 \rceil) + cN \\
\leq 20c \lceil N/5 \rceil + 20c( N - 3\lceil N/10 \rceil ) + cN \\
\leq 20c(N/5) + 20c(N) - 20c(N/4) + cN \\
= 20cN
\]

\[
\text{For } n \geq 50, \text{ } 3\lceil N/10 \rceil \geq N/4.
\]

Linear Time Selection Postmortem

Practical considerations.
- Constant (currently) too large to be useful.
- Practical variant: choose random partition element.
  - $O(N)$ expected running time ala quicksort.
- Open problem: guaranteed $O(N)$ with better constant.

Quicksort.
- Worst case $O(N \log N)$ if always partition on median.