Deterministic Selection

How to find a "good" splitter, so that at least a constant fraction, say \( \frac{n}{4} \) elements are eliminated.

1. Sort every column containing at most \( \frac{n}{5} \) elements.
   Time: \( O(1) \cdot \frac{n}{5} = O(n) \)

2. Find the "median" of the median of each row, i.e., median of \( \frac{n}{5} \) elements.

3. Find the rank of \( M \) \( O(n) \)
Claim: The median of medians has rank in the range $[\alpha n, (1-\alpha)n]$ for some constant $\alpha < 1$.

\[ \frac{n}{10} \times 3 \text{ elements are smaller than } M \]

\[ \frac{3n}{10} \text{ } \Rightarrow \text{ larger than } M \]

\[ \alpha = \frac{3}{10} \]

Step 4: If $\text{rank}(M, S) < k$
then we can discard all elements in the shaded box to the right bottom.

$\text{Time} : O(n)$

Steps 5: Call recursively on the remaining elements with the adjusted value of $k$.

$T(n) : \text{the worst case time over all } k < n$

\[ T(n) = T\left(\frac{n}{5}\right) + O(n) \quad T(2) : O(1) \]
\[ + T\left(\frac{3}{10}n\right) \]
Prove (using induction) that
\[ T(n) \leq \alpha \cdot n \] for some constant \( \alpha \).

**Soft-heaps** by Chazelle

Prune and Search
Finite alphabet \( \Sigma \)

Strings \( s_1, s_2, \ldots, s_n \) defined over \( \Sigma \) of lengths \( l_1, l_2, \ldots \)

\[ |s_i| = l_i \]

\( \sigma_1, 1100, 101010, \ldots \)

\( L = \Sigma l_i \)

Possible that some string has size \( 1 \)

Whereas some string may have size \( \frac{L}{4} \)

We would like to order the strings use "lexicographic ordering"

What algorithm should we use?

Radix sort?

100, 255, 099, 005, \ldots, 855
Using count sort, time for each phase \( \Theta(|\Xi| + n) \)

Stable sort: doesn't alter the ordering of numbers with identical values.

Radix sort has to be based on some stable sorting.

Count Sort: Stable?

Every comparison sorting can be made stable.
Radar Sort

$O(d(\log d))$

as long as $d$ is constant

$\Sigma = 1..n$, i.e. logn bits

We can sort $n$ numbers in the range $[1..nd]$ in $O(n)$ time.