CS 381

Deterministic Selection Algorithms

Goal: Find k-th smallest element of an array of n elements.

Deterministic Selection

Note: This was thought to be impossible for a long time (to do efficiently), until a 1973 paper, by Blum, Floyd, Pratt, Rivest and Tarjan proposed the “median-of-medians” algorithm.

Deterministic Selection

1. Divide the elements into **groups of five**, where the last group may have less than five elements in case when the input array size is not a multiple of five.

2. Compute the **median** of each group (ties can be broken arbitrarily).

3. Make a recursive call to calculate the **median of the medians**. Set x to the median.

4. Use x as the pivot and partition.

5. If the pivot is not the order statistics that is searched for, recurse on the subarray that contains it.

Deterministic Selection

Use a bound B to stop recursion: If the size of the array is less than or equal to B then use brute-force search to find the desired order statics.

Pseudocode

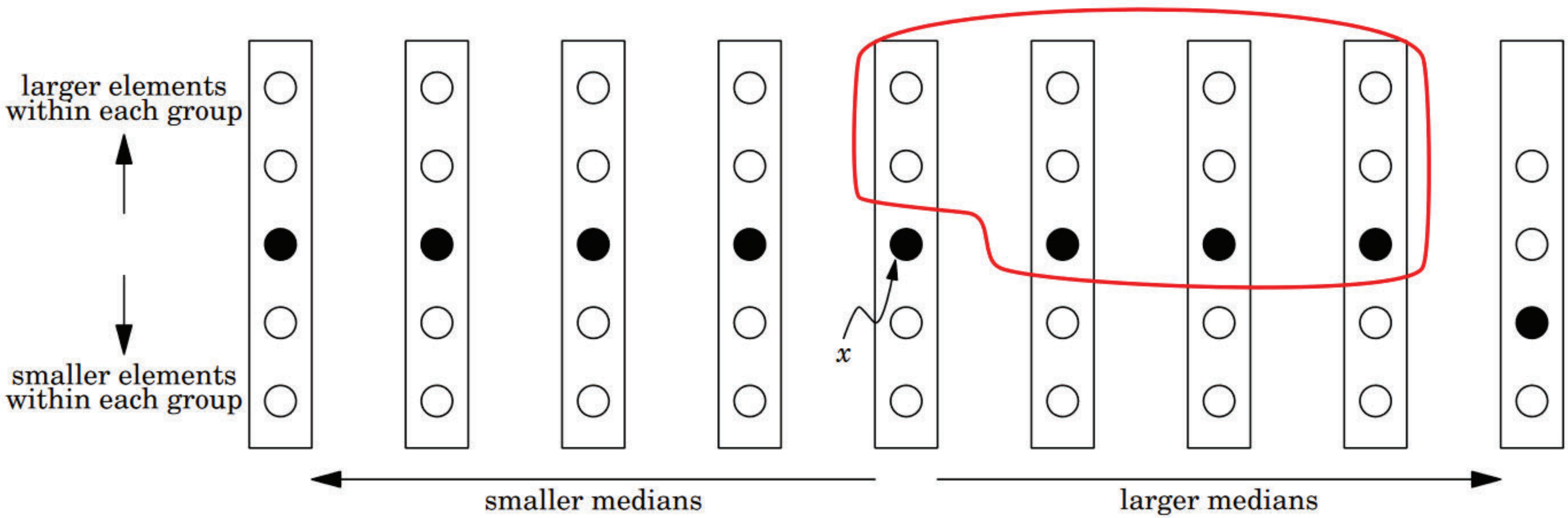
Algorithm: $\textsc{Select}(A, i)$

1. Divide the n items into groups of 5 (plus any remainder).
2. Find the median of each group of 5 (by rote). (If the remainder group has an even number of elements, then break ties arbitrarily, for example by choosing the lower median.)
3. Use $\textsc{Select}$ recursively to find the median (call it x) of these $\lceil n/5 \rceil$ medians.
4. Partition around x.[*] Let $k = \text{rank}(x)$.[†]

$$\underbrace{\boxed{\quad A_i < x \quad}}_{k-1 \text{ elements}}\underset{\substack{\uparrow \\ A_k}}{\boxed{x}}\underbrace{\boxed{\quad A_i > x \quad}}_{n-k \text{ elements}}$$

5.
   - If $i = k$, then return x.
   - Else, if $i < k$, use $\textsc{Select}$ recursively by calling $\textsc{Select}(A[1,\ldots,k-1],\ i)$.[‡]
   - Else, if $i > k$, use $\textsc{Select}$ recursively by calling $\textsc{Select}(A[k+1,\ldots,i],\ i-k)$.

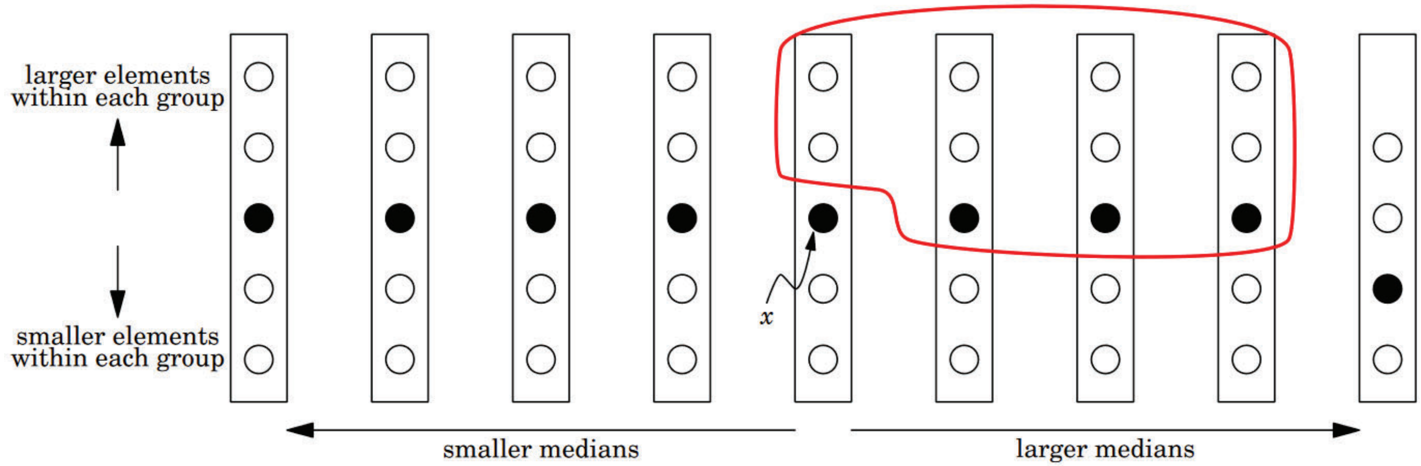
Imagine laying out the groups side by side, ordered with respect to their medians. Also imagine each group is sorted from greatest to least, top to bottom.



x : median of medians

Q: How large can the sub-array be after running Partition?

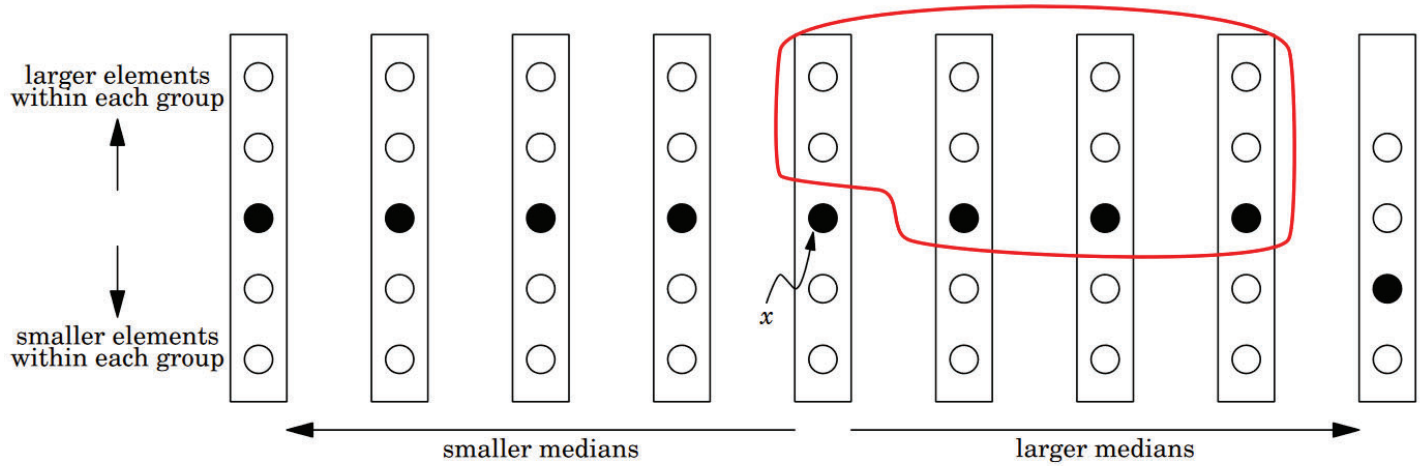
Deriving bounds on the subarrays



Each group of size 5 with median

- at most x contains at least 3 elements smaller than x.
- at least x contains at least 3 elements greater than x.

Deriving bounds on the subarrays



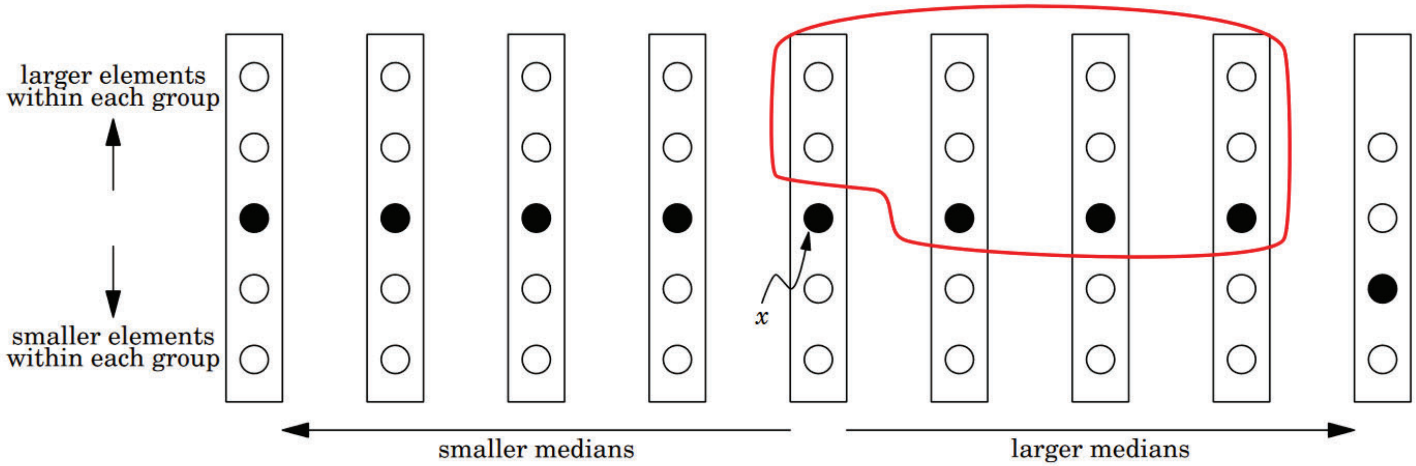
Each group of size 5 with median

- at most x contains at least 3 elements smaller than x.
- at least x contains at least 3 elements greater than x.

There are $\geq \lceil n/5 \rceil - 1$ groups of size 5:

- at least $\lfloor \left(\frac{1}{2} \cdot \lceil n/5 \rceil - 2\right) \rfloor$ of them have median smaller than x
- at least $\lfloor \left(\frac{1}{2} \cdot \lceil n/5 \rceil - 2\right) \rfloor$ of them have median larger than x.

Deriving bounds on the subarrays



Each group of size 5 with median

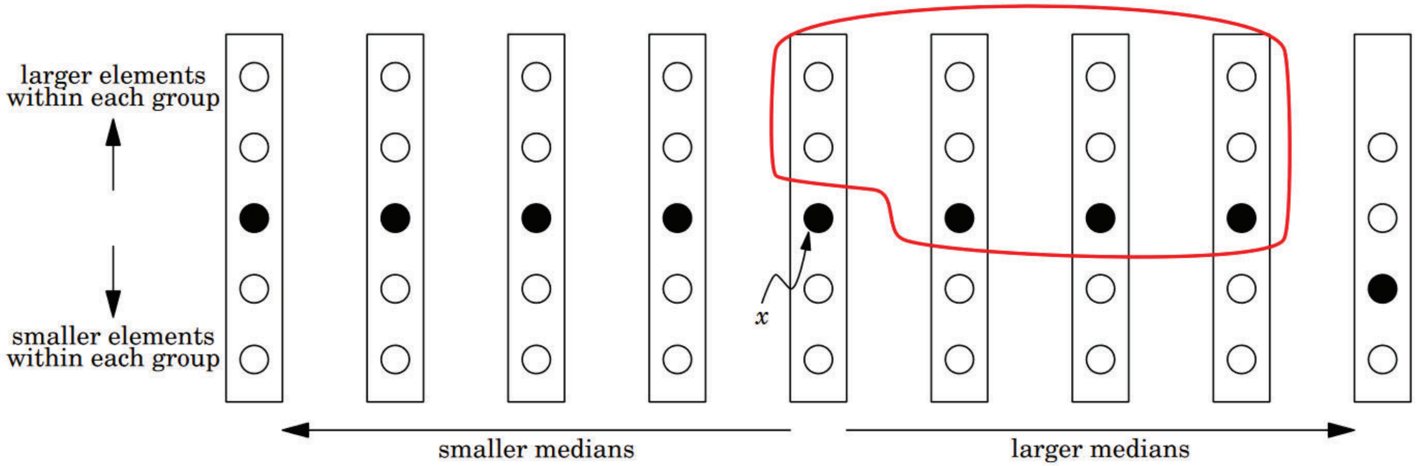
- at most x contains at least 3 elements smaller than x.
- at least x contains at least 3 elements greater than x.

There are $\geq \lceil n/5 \rceil - 1$ groups of size 5:

- at least $\lfloor \left(\frac{1}{2} \cdot \lceil n/5 \rceil - 2\right) \rfloor$ of them have median smaller than x
- at least $\lfloor \left(\frac{1}{2} \cdot \lceil n/5 \rceil - 2\right) \rfloor$ of them have median larger than x.

The group with x as median has 2 elements greater than x and 2 smaller than x

Deriving bounds on the subarrays

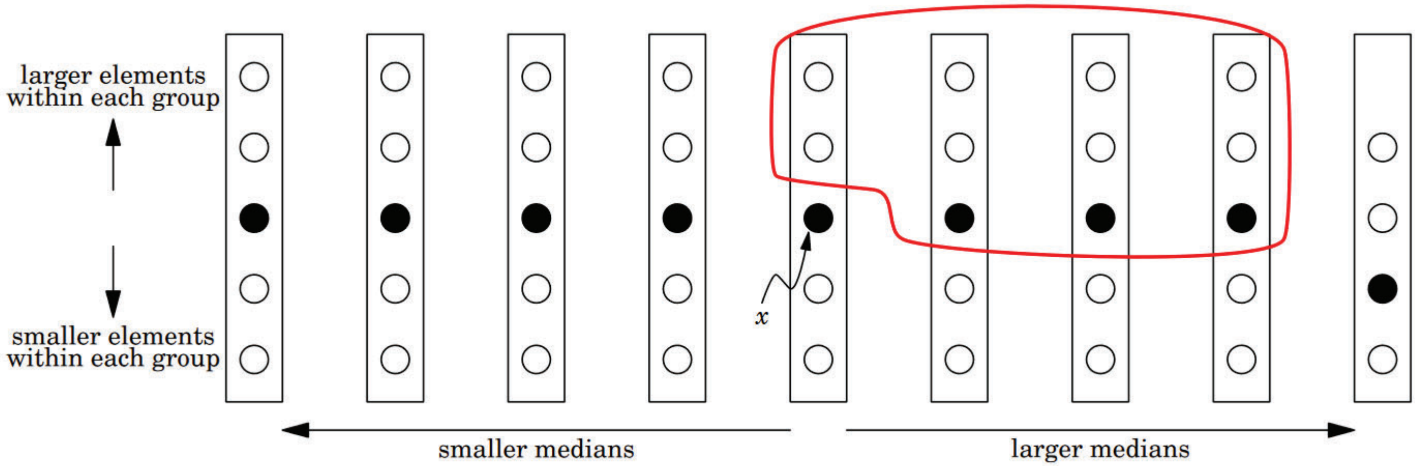


Combining these observations, we get:

$$(\#\text{ elts of } A \text{ less than } x) \geq 3\left(\left\lfloor \tfrac{1}{2}\lceil n/5\rceil - 2\right\rfloor\right) + 2 \geq 3\left(\tfrac{1}{2}\lceil n/5\rceil - 2 - \tfrac{1}{2}\right) + 2 > \tfrac{3}{10}n - 6.$$

$$(\#\text{ elts of } A \text{ greater than } x) \geq \tfrac{3}{10}n - 6.$$

Deriving bounds on the subarrays



Combining these observations, we get:

$$(\#\text{ elts of } A \text{ less than } x) \geq 3\left(\left\lfloor \frac{1}{2}\lceil n/5 \rceil - 2 \right\rfloor\right) + 2 \geq 3\left(\frac{1}{2}\lceil n/5 \rceil - 2 - \frac{1}{2}\right) + 2 > \frac{3}{10}n - 6.$$

$$(\#\text{ elts of } A \text{ greater than } x) \geq \frac{3}{10}n - 6.$$

Lemma: The larger subarray has at most $\frac{7}{10}n + 6$ elements.

Runtime Analysis

Assume the input numbers are pairwise distinct.

Let $T(n)$ be the max number of comparisons of this method, where the max is taken over the worst case index k of the statistic searched for.

Runtime Analysis

Assume the input numbers are pairwise distinct.

Let $T(n)$ be the max number of comparisons of this method, where the max is taken over the worst case index k of the statistic searched for.

Claim: there is a constant α such that $T(n) \leq \alpha \cdot n$ for all $n \geq 1$.

Runtime Analysis

Assume the input numbers are pairwise distinct.

Let $T(n)$ be the max number of comparisons of this method, where the max is taken over the worst case index k of the statistic searched for.

Claim: there is a constant α such that $T(n) \leq \alpha \cdot n$ for all $n \geq 1$.

Recall B is the bound to stop the recursion: if the array has size $\leq$ B, compute the median by brute force.

Runtime Analysis

Assume the input numbers are pairwise distinct.

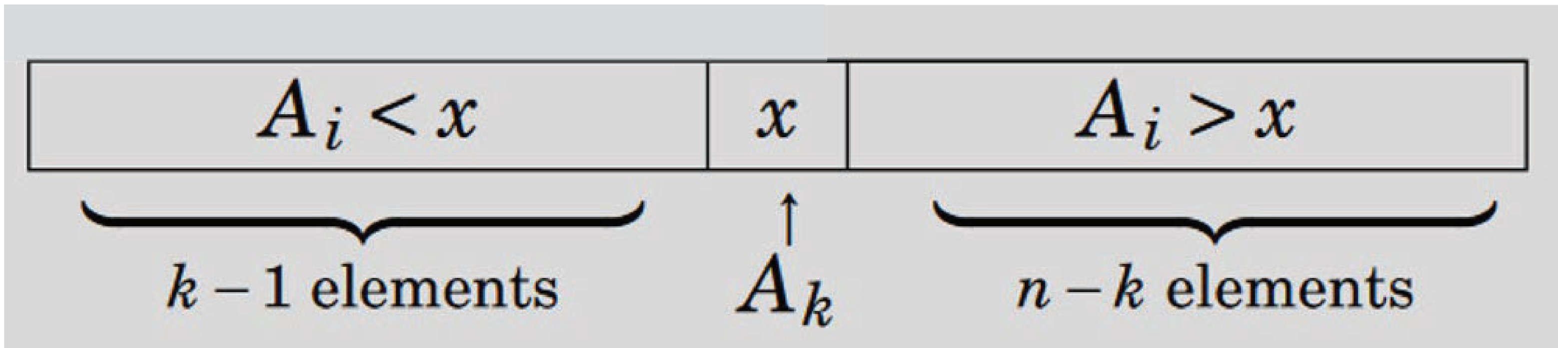
Let $T(n)$ be the max number of comparisons of this method, where the max is taken over the worst case index k of the statistic searched for.

Claim: there is a constant α such that $T(n) \leq \alpha \cdot n$ for all $n \geq 1$.

Recall B is the bound to stop the recursion: if the array has size $\leq$ B, compute the median by brute force.

As long as B is set to a constant, we can adjust a value of α so that the claim holds for all $n \leq B$.

To show the claim when $n \geq B$:



By previous **Lemma**, the larger subarray has at most $\frac{7}{10}n + 6$ elements.

Let β be a constant so that the partition operation and division into groups of five takes at most βn comparisons.

Q: How can we upper bound $T(n)$?

Pseudocode

Algorithm: SELECT(A, i)

1. Divide the n items into groups of 5 (plus any remainder).

2. Find the median of each group of 5 (by rote). (If the remainder group has an even number of elements, then break ties arbitrarily, for example by choosing the lower median.)

3. Use SELECT recursively to find the median (call it x) of these $\lceil n/5 \rceil$ medians.

4. Partition around x.[*] Let $k = \text{rank}(x)$.[†]

$$\underbrace{\boxed{\quad A_i < x \quad}}_{k-1 \text{ elements}}\ \underset{\substack{\uparrow \\ A_k}}{\boxed{x}}\ \underbrace{\boxed{\quad A_i > x \quad}}_{n-k \text{ elements}}$$

5.
   - If $i = k$, then return x.
   - Else, if $i < k$, use SELECT recursively by calling SELECT$(A[1, \ldots, k-1],\ i)$.[‡]
   - Else, if $i > k$, use SELECT recursively by calling SELECT$(A[k+1, \ldots, i],\ i-k)$.

T(n) satisfies the inequality

$$T(n) \leq \beta n + T\left(\frac{n}{5} + 1\right) + T\left(\frac{7n}{10} + 6\right)$$

T(n) satisfies the inequality

$$T(n) \leq \beta n + T\left(\frac{n}{5} + 1\right) + T\left(\frac{7n}{10} + 6\right)$$

By applying the induction hypothesis we get:

$$T(n) \leq \beta n + \alpha\left(\frac{n}{5} + 1 + \frac{7n}{10} + 6\right)$$

This is

$$T(n) \le \beta n + \frac{9\alpha}{10} n + 7\alpha$$

$$T(n) \le \alpha n + \beta n - \frac{\alpha}{10} n + 7\alpha$$

which is $\le \alpha n$ if

$$\beta n - \frac{\alpha}{10} n + 7\alpha \le 0$$

$$\beta n - \frac{\alpha}{10} n + 7\alpha \leq 0$$

$$-10\beta n + (n - 70)\alpha \geq 0$$

$$\alpha \geq 10\beta \frac{n}{n - 70}$$

Let $B = 140$, choose $\alpha \geq 20\beta$ to show $T(n) \leq \alpha n$.