Algorithms

Runtime Analysis And Order Statistics

Dr. Mudassir Shabbir

LUMS University

January 22, 2026

Selection Problem

Problem

Given an array A of n distinct numbers, find the k-th smallest element.

- Sorting gives $O(n \log n)$ time
- Can we do better?
- **Yes:** Deterministic linear-time selection

Median of Medians Algorithm

High-Level Idea

- Use a **good pivot** for partitioning
- Pivot should be guaranteed to be "near the middle"
- Avoid worst-case $O(n^2)$ behavior

High-Level Idea

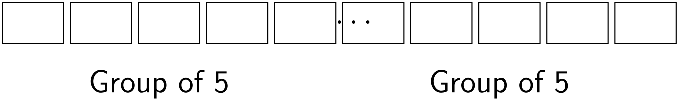
- Use a **good pivot** for partitioning
- Pivot should be guaranteed to be "near the middle"
- Avoid worst-case $O(n^2)$ behavior

Key Insight

We can find a pivot whose rank is guaranteed to be between $\frac{3n}{10}$ and $\frac{7n}{10}$.

Step 1: Grouping

- Divide the array into groups of 5
- Each group has at most 5 elements



Step 2: Medians of Groups

- Sort each group of 5 (constant time per group)
- Extract the median of each group

Step 2: Medians of Groups

- Sort each group of 5 (constant time per group)
- Extract the median of each group

Observation

Number of medians $= \lceil n/5 \rceil$

Step 2: Medians of Groups

- Sort each group of 5 (constant time per group)
- Extract the median of each group

Observation

Number of medians $= \lceil n/5 \rceil$

- Recursively find the median of these medians
- This value is called the **median of medians**

Step 3: Partitioning

- Use the median of medians as a pivot
- Partition the array around this pivot

Step 3: Partitioning

- Use the median of medians as a pivot
- Partition the array around this pivot

Guarantee

At least $\frac{3n}{10}$ elements are $\leq$ pivot At least $\frac{3n}{10}$ elements are $\geq$ pivot

Step 3: Partitioning

- Use the median of medians as a pivot
- Partition the array around this pivot

Guarantee

At least $\frac{3n}{10}$ elements are $\leq$ pivot At least $\frac{3n}{10}$ elements are $\geq$ pivot

- Recurse only on the relevant side

Resulting Recurrence

- One recursive call to find median of medians:

$$T(n/5)$$

- One recursive call after partitioning:

$$T(7n/10)$$

- Linear work for grouping, sorting, partitioning

Resulting Recurrence

- One recursive call to find median of medians:

$$T(n/5)$$

- One recursive call after partitioning:

$$T(7n/10)$$

- Linear work for grouping, sorting, partitioning

Recurrence

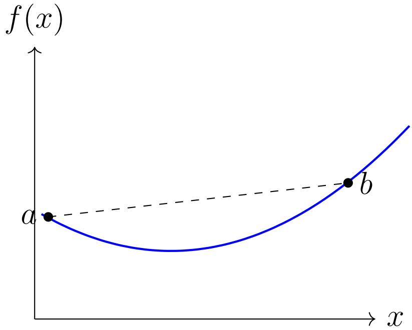
$$T(n) \leq T(n/5) + T(7n/10) + cn$$

Convex Functions

Definition

A function f is **convex** if for all a, b and $\alpha \in [0,1]$,

$$f(\alpha a + (1-\alpha)b) \leq \alpha f(a) + (1-\alpha) f(b)$$



A Useful Property

Assumption

Assume f is convex and $f(0) = 0$

A Useful Property

Assumption

Assume f is convex and $f(0) = 0$

Claim

For $0 \leq \alpha \leq 1$,

$$f(\alpha a) \leq \alpha f(a)$$

A Useful Property

Assumption

Assume f is convex and $f(0) = 0$

Claim

For $0 \leq \alpha \leq 1$,

$$f(\alpha a) \leq \alpha f(a)$$

- Apply convexity with $b = 0$
- Use $f(0) = 0$

Subadditivity of Convex Functions

Claim

For all $x, y \geq 0$,

$$f(x) + f(y) \leq f(x + y)$$

Subadditivity of Convex Functions

Claim

For all $x, y \geq 0$,

$$f(x) + f(y) \leq f(x+y)$$

- Let $\alpha = \frac{x}{x+y}$
- Use $f(\alpha(x+y)) \leq \alpha f(x+y)$
- Rearranging gives the result

Applying This to the Recurrence

- Let $f(n) = T(n)$
- Assume T is non-decreasing and convex

Applying This to the Recurrence

- Let $f(n) = T(n)$
- Assume T is non-decreasing and convex

$$T(n/5) + T(7n/10) \leq T\left(\frac{n}{5} + \frac{7n}{10}\right) = T\left(\frac{9n}{10}\right)$$

Applying This to the Recurrence

- Let $f(n) = T(n)$
- Assume T is non-decreasing and convex

$$T(n/5) + T(7n/10) \leq T\left(\frac{n}{5} + \frac{7n}{10}\right) = T\left(\frac{9n}{10}\right)$$

$$T(n) \leq T(9n/10) + cn$$

Solving the Final Recurrence

Final Recurrence

$$T(n) \leq T(9n/10) + cn$$

Solving the Final Recurrence

Final Recurrence

$$T(n) \leq T(9n/10) + cn$$

- Recursion depth: $O(\log n)$
- Total work:

$$cn + c(9n/10) + c(9^2 n/10^2) + \cdots$$

Solving the Final Recurrence

Final Recurrence

$$T(n) \leq T(9n/10) + cn$$

- Recursion depth: $O(\log n)$
- Total work:

$$cn + c(9n/10) + c(9^2 n/10^2) + \cdots$$

$$T(n) = O(n)$$

Takeaway

- Median of Medians gives a **deterministic linear-time** selection algorithm
- Careful pivot choice guarantees balanced recursion
- Convexity provides a clean way to reason about the recurrence

Worst-case linear time is achievable!