Algorithms and datastructures I
Lecture 11: Master theorem, Strassen algorithm, k-th smallest element

Jan Hubička

Department of Applied Mathematics
Charles University
Prague

April 28 2020

Divide & Conquer

“Divide and conquer is an algorithm design paradigm based on multi-branched recursion. A divide-and-conquer algorithm works by recursively breaking down a problem into two or more sub-problems of the same or related type, until these become simple enough to be solved directly. The solutions to the sub-problems are then combined to give a solution to the original problem.”

John von Neumann

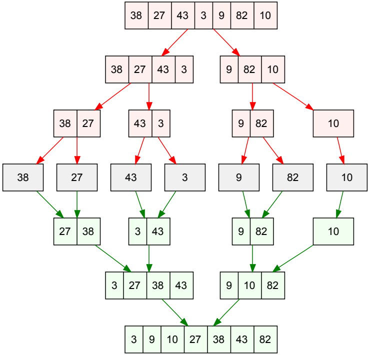
MergeSort $(x_1, \ldots, x_n)$, John von Neumann, 1945

1. if $n = 1$: Return (x_1).
2. $(y_1, \ldots, y_{\lfloor \frac{n}{2} \rfloor}) \leftarrow$MergeSort $(x_1, \ldots, x_{\lfloor \frac{n}{2} \rfloor})$
3. $(z_1, \ldots, z_{\lceil \frac{n}{2} \rceil}) \leftarrow$MergeSort $(x_{\lfloor \frac{n}{2} \rfloor + 1}, \ldots, x_n)$
4. Return Merge $((y_1, \ldots, y_{\lfloor \frac{n}{2} \rfloor}), (z_1, \ldots, z_{\lceil \frac{n}{2} \rceil}))$.

Time complexity (for $n = 2^k$)

$$T(n) = 2T(\frac{n}{2}) + \Theta(n)$$

$$T(1) = 1$$



Multiplication (Karatsuba 1960)

$$
\begin{aligned}
X &= \boxed{A}\boxed{B} = A \cdot 10^{\frac{n}{2}} + B \\
Y &= \boxed{C}\boxed{D} = C \cdot 10^{\frac{n}{2}} + D \\
X \cdot Y &= AC \cdot 10^n + (AD + BC) \cdot 10^{\frac{n}{2}} + BD \\
&= AC \cdot 10^n + ((A+B)(C+D) - AC - BD) \cdot 10^{\frac{n}{2}} + BD
\end{aligned}
$$

Anatolii Alexeievitch Karatsuba

$$T(n) = 3T(\frac{n}{2}) + cn$$

$$
\begin{aligned}
T(n) &= \sum_{i=0}^{\log n} \left(\frac{3}{2}\right)^i cn = cn \cdot \frac{\left(\frac{3}{2}\right)^{\log n + 1} - 1}{\left(\frac{3}{2}\right) - 1} = \Theta\left(n \cdot \left(\frac{3}{2}\right)^{\log n}\right) = \Theta\left(n \cdot \left(\frac{3^{\log n}}{2^{\log n}}\right)\right) = \Theta(3^{\log n}) \\
&= \Theta\left(\left(2^{\log 3}\right)^{\log n}\right) = \Theta\left(2^{\log 3 \log n}\right) = \Theta\left(\left(2^{\log n}\right)^{\log 3}\right) = \Theta\left(n^{\log 3}\right) = \Theta\left(n^{1.59\cdots}\right).
\end{aligned}
$$

What about general case?

General recurrence

$$
\begin{aligned}
T(1) &= 1 \\
T(n) &= aT(\lfloor \frac{n}{b} \rfloor) + \Theta(n^c)
\end{aligned}
$$

What about general case?

General recurrence

$$
\begin{aligned}
T(1) &= 1 \\
T(n) &= aT(\lfloor \frac{n}{b} \rfloor) + \Theta(n^c)
\end{aligned}
$$

# of subprob	size of subprob.	time per subprob	time per level

Lets do the math

$$T(N) = \Theta\left(\sum_{i=0}^{k} N^c \frac{a^i}{b^{ic}}\right)$$

Lets do the math

$$T(N) = \Theta\left(\sum_{i=0}^{k} N^c \frac{a^i}{b^{ic}}\right) = \Theta\left(N^c \sum_{i=0}^{k} \left(\frac{a}{b^c}\right)^i\right)$$

Lets do the math

$$T(N) = \Theta\left(\sum_{i=0}^{k} N^c \frac{a^i}{b^{ic}}\right) = \Theta\left(N^c \sum_{i=0}^{k} \left(\frac{a}{b^c}\right)^i\right)$$

Put $q = \frac{a}{b^c}$ and consider cases:

1. $q = 1$: $T(n) = \Theta\left(n^c \log n\right)$

Lets do the math

$$T(N) = \Theta\left(\sum_{i=0}^{k} N^c \frac{a^i}{b^{ic}}\right) = \Theta\left(N^c \sum_{i=0}^{k} \left(\frac{a}{b^c}\right)^i\right)$$

Put $q = \frac{a}{b^c}$ and consider cases:

1. $q = 1$: $T(n) = \Theta\left(n^c \log n\right)$
2. $q < 1$: $T(n) = \Theta\left(n^c \sum_{i \geq 0} q^i\right)$

Lets do the math

$$T(N) = \Theta\left(\sum_{i=0}^{k} N^c \frac{a^i}{b^{ic}}\right) = \Theta\left(N^c \sum_{i=0}^{k} \left(\frac{a}{b^c}\right)^i\right)$$

Put $q = \frac{a}{b^c}$ and consider cases:

1. $q = 1$: $T(n) = \Theta\left(n^c \log n\right)$
2. $q < 1$: $T(n) = \Theta\left(n^c \sum_{i \geq 0} q^i\right) = \Theta\left(n^c\right)$

Lets do the math

$$T(N) = \Theta\left(\sum_{i=0}^{k} N^c \frac{a^i}{b^{ic}}\right) = \Theta\left(N^c \sum_{i=0}^{k} \left(\frac{a}{b^c}\right)^i\right)$$

Put $q = \frac{a}{b^c}$ and consider cases:

1. $q = 1$: $T(n) = \Theta\left(n^c \log n\right)$
2. $q < 1$: $T(n) = \Theta\left(n^c \sum_{i \geq 0} q^i\right) = \Theta\left(n^c\right)$
3. $q > 1$: $T(n) = \Theta\left(a^k\right)$

Lets do the math

$$T(N) = \Theta\left(\sum_{i=0}^{k} N^c \frac{a^i}{b^{ic}}\right) = \Theta\left(N^c \sum_{i=0}^{k} \left(\frac{a}{b^c}\right)^i\right)$$

Put $q = \frac{a}{b^c}$ and consider cases:

1. $q = 1$: $T(n) = \Theta\left(n^c \log n\right)$
2. $q < 1$: $T(n) = \Theta\left(n^c \sum_{i \geq 0} q^i\right) = \Theta\left(n^c\right)$
3. $q > 1$: $T(n) = \Theta\left(a^k\right) = \Theta\left(a^{\log_b n}\right)$

Lets do the math

$$T(N) = \Theta\left(\sum_{i=0}^{k} N^c \frac{a^i}{b^{ic}}\right) = \Theta\left(N^c \sum_{i=0}^{k} \left(\frac{a}{b^c}\right)^i\right)$$

Put $q = \frac{a}{b^c}$ and consider cases:

1. $q = 1$: $T(n) = \Theta\left(n^c \log n\right)$
2. $q < 1$: $T(n) = \Theta\left(n^c \sum_{i \geq 0} q^i\right) = \Theta\left(n^c\right)$
3. $q > 1$: $T(n) = \Theta\left(a^k\right) = \Theta\left(a^{\log_b n}\right) = \Theta\left(\left(b^{\log_b a}\right)^{\log_b n}\right)$

Lets do the math

$$T(N) = \Theta\left(\sum_{i=0}^{k} N^c \frac{a^i}{b^{ic}}\right) = \Theta\left(N^c \sum_{i=0}^{k} \left(\frac{a}{b^c}\right)^i\right)$$

Put $q = \frac{a}{b^c}$ and consider cases:

1. $q = 1$: $T(n) = \Theta\left(n^c \log n\right)$
2. $q < 1$: $T(n) = \Theta\left(n^c \sum_{i \geq 0} q^i\right) = \Theta\left(n^c\right)$
3. $q > 1$: $T(n) = \Theta\left(a^k\right) = \Theta\left(a^{\log_b n}\right) = \Theta\left(\left(b^{\log_b a}\right)^{\log_b n}\right) = \Theta\left(\left(b^{\log_b n}\right)^{\log_b a}\right)$

Lets do the math

$$T(N) = \Theta\left(\sum_{i=0}^{k} N^c \frac{a^i}{b^{ic}}\right) = \Theta\left(N^c \sum_{i=0}^{k} \left(\frac{a}{b^c}\right)^i\right)$$

Put $q = \frac{a}{b^c}$ and consider cases:

1. $q = 1$: $T(n) = \Theta\left(n^c \log n\right)$
2. $q < 1$: $T(n) = \Theta\left(n^c \sum_{i \geq 0} q^i\right) = \Theta\left(n^c\right)$
3. $q > 1$: $T(n) = \Theta\left(a^k\right) = \Theta\left(a^{\log_b n}\right) = \Theta\left(\left(b^{\log_b a}\right)^{\log_b n}\right) = \Theta\left(\left(b^{\log_b n}\right)^{\log_b a}\right) = \Theta\left(n^{\log_b a}\right)$

Lets do the math

$$T(N) = \Theta\left(\sum_{i=0}^{k} N^c \frac{a^i}{b^{ic}}\right) = \Theta\left(N^c \sum_{i=0}^{k} \left(\frac{a}{b^c}\right)^i\right)$$

Put $q = \frac{a}{b^c}$ and consider cases:

1. $q = 1$: $T(n) = \Theta\left(n^c \log n\right)$
2. $q < 1$: $T(n) = \Theta\left(n^c \sum_{i \geq 0} q^i\right) = \Theta\left(n^c\right)$
3. $q > 1$: $T(n) = \Theta\left(a^k\right) = \Theta\left(a^{\log_b n}\right) = \Theta\left(\left(b^{\log_b a}\right)^{\log_b n}\right) = \Theta\left(\left(b^{\log_b n}\right)^{\log_b a}\right) = \Theta\left(n^{\log_b a}\right)$

Question

What if $n \leq b^k$ for some integer k?

Lets do the math

$$T(N) = \Theta\left(\sum_{i=0}^{k} N^c \frac{a^i}{b^{ic}}\right) = \Theta\left(N^c \sum_{i=0}^{k} \left(\frac{a}{b^c}\right)^i\right)$$

Put $q = \frac{a}{b^c}$ and consider cases:

1. $q = 1$: $T(n) = \Theta\left(n^c \log n\right)$
2. $q < 1$: $T(n) = \Theta\left(n^c \sum_{i \geq 0} q^i\right) = \Theta\left(n^c\right)$
3. $q > 1$: $T(n) = \Theta\left(a^k\right) = \Theta\left(a^{\log_b n}\right) = \Theta\left(\left(b^{\log_b a}\right)^{\log_b n}\right) = \Theta\left(\left(b^{\log_b n}\right)^{\log_b a}\right) = \Theta\left(n^{\log_b a}\right)$

Question

What if $n \leq b^k$ for some integer k?

Easy: Put $b^k \leq n \leq b^{k+1}$ and then $T(b^k) \leq T(n) \leq T(b^{k+1})$

Master theorem

Theorem (Master theorem)

Master theorem

Theorem (Master theorem)

Given $a \in \mathbb{N}^+, b \geq 1, c \geq 1$ *recurrence:*

$$\begin{aligned} T(1) &= 1 \\ T(n) &= aT(\lfloor \frac{n}{b} \rfloor) + \Theta(n^c) \end{aligned}$$

has solution:

1. $T(n) = \Theta\left(n^c \log n\right)$ *if* $\frac{a}{b^c} = 1$.
2. $T(n) = \Theta\left(n^c\right)$ *if* $\frac{a}{b^c} < 1$.
3. $T(n) = \Theta\left(n^{\log_b a}\right)$ *if* $\frac{a}{b^c} > 1$.

Master theorem

Theorem (Master theorem)

Given $a \in \mathbb{N}^+, b \geq 1, c \geq 1$ *recurrence:*

$$\begin{aligned} T(1) &= 1 \\ T(n) &= aT(\lfloor \frac{n}{b} \rfloor) + \Theta(n^c) \end{aligned}$$

has solution:

1. $T(n) = \Theta\left(n^c \log n\right)$ *if* $\frac{a}{b^c} = 1$.
2. $T(n) = \Theta\left(n^c\right)$ *if* $\frac{a}{b^c} < 1$.
3. $T(n) = \Theta\left(n^{\log_b a}\right)$ *if* $\frac{a}{b^c} > 1$.

Strassen's algorithm

Volker Strassen

Strassen's algorithm, 1969

$$\begin{pmatrix} A & B \\ C & D \end{pmatrix} \cdot \begin{pmatrix} P & Q \\ R & S \end{pmatrix}$$

Strassen's algorithm

Volker Strassen

Strassen's algorithm, 1969

$$\begin{pmatrix} A & B \\ C & D \end{pmatrix} \cdot \begin{pmatrix} P & Q \\ R & S \end{pmatrix} = \begin{pmatrix} T_1 + T_4 - T_5 + T_7 & T_3 + T_5 \\ T_2 + T_4 & T_1 - T_2 + T_3 + T_6 \end{pmatrix},$$

where:

$$\begin{aligned} T_1 &= (A + D) \cdot (P + S) \\ T_2 &= (C + D) \cdot P \\ T_3 &= A \cdot (Q - S) \\ T_4 &= D \cdot (R - P) \\ T_5 &= (A + B) \cdot S \\ T_6 &= (C - A) \cdot (P + Q) \\ T_7 &= (B - D) \cdot (R + S) \end{aligned}$$

Strassen's algorithm

Volker Strassen

Strassen's algorithm, 1969

$$\begin{pmatrix} A & B \\ C & D \end{pmatrix} \cdot \begin{pmatrix} P & Q \\ R & S \end{pmatrix} = \begin{pmatrix} T_1 + T_4 - T_5 + T_7 & T_3 + T_5 \\ T_2 + T_4 & T_1 - T_2 + T_3 + T_6 \end{pmatrix},$$

where:

$$\begin{aligned} T_1 &= (A + D) \cdot (P + S) \\ T_2 &= (C + D) \cdot P \\ T_3 &= A \cdot (Q - S) \\ T_4 &= D \cdot (R - P) \\ T_5 &= (A + B) \cdot S \\ T_6 &= (C - A) \cdot (P + Q) \\ T_7 &= (B - D) \cdot (R + S) \end{aligned}$$

7 multiplications instead of 8 $\Rightarrow$ time complexity $T(n) = 7T\left(\frac{n}{2}\right) + \Theta(n^2) = \Theta(n^{\log_2 7}) = O(n^{2.808})$.

Strassen's algorithm

Volker Strassen

Strassen's algorithm, 1969

$$\begin{pmatrix} A & B \\ C & D \end{pmatrix} \cdot \begin{pmatrix} P & Q \\ R & S \end{pmatrix} = \begin{pmatrix} T_1 + T_4 - T_5 + T_7 & T_3 + T_5 \\ T_2 + T_4 & T_1 - T_2 + T_3 + T_6 \end{pmatrix},$$

where:

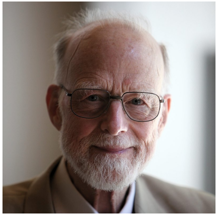
$$\begin{aligned} T_1 &= (A + D) \cdot (P + S) \\ T_2 &= (C + D) \cdot P \\ T_3 &= A \cdot (Q - S) \\ T_4 &= D \cdot (R - P) \\ T_5 &= (A + B) \cdot S \\ T_6 &= (C - A) \cdot (P + Q) \\ T_7 &= (B - D) \cdot (R + S) \end{aligned}$$

7 multiplications instead of 8 $\Rightarrow$ time complexity $T(n) = 7T\left(\frac{n}{2}\right) + \Theta(n^2) = \Theta(n^{\log_2 7}) = O(n^{2.808})$.

Current record: $(n^{2.373})$ with really big constant factors

Quickselect

Problem: Find k-th smallest element of a sequence $(x_1, \dots, x_n)$.



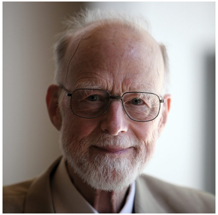
Sir Tony Horae

QuickSelect($(x_1, \dots, x_n)$, k), Sir Tony Horae 1961

1. Choose pivot p.
2. Split $(x_1, \dots, x_n)$ to $L = \{x_i : x_i < p\}$, $E = \{x_i : x_i = p\}$, $R = \{x_i : x_i > p\}$.

Quickselect

Problem: Find k-th smallest element of a sequence $(x_1, \ldots, x_n)$.



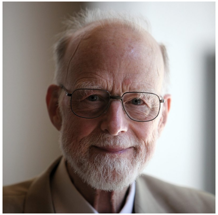
Sir Tony Horae

QuickSelect($(x_1, \ldots, x_n)$, k), Sir Tony Horae 1961

1. Choose pivot p.
2. Split $(x_1, \ldots, x_n)$ to $L = \{x_i : x_i < p\}$, $E = \{x_i : x_i = p\}$, $R = \{x_i : x_i > p\}$.
3. If $k \leq |L|$: return QuickSelect(L,k).

Quickselect

Problem: Find k-th smallest element of a sequence $(x_1, \ldots, x_n)$.



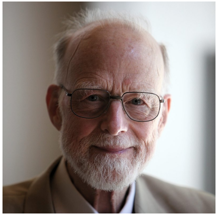
Sir Tony Horae

QuickSelect($(x_1, \ldots, x_n)$, k), Sir Tony Horae 1961

1. Choose pivot p.
2. Split $(x_1, \ldots, x_n)$ to $L = \{x_i : x_i < p\}$, $E = \{x_i : x_i = p\}$, $R = \{x_i : x_i > p\}$.
3. If $k \leq |L|$: return QuickSelect(L,k).
4. If $k \leq |L| + |E|$: return k.

Quickselect

Problem: Find k-th smallest element of a sequence $(x_1, \ldots, x_n)$.



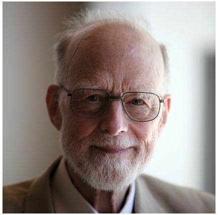
Sir Tony Horae

QuickSelect($(x_1, \ldots, x_n)$, k), Sir Tony Horae 1961

1. Choose pivot p.
2. Split $(x_1, \ldots, x_n)$ to $L = \{x_i : x_i < p\}$, $E = \{x_i : x_i = p\}$, $R = \{x_i : x_i > p\}$.
3. If $k \leq |L|$: return QuickSelect(L,k).
4. If $k \leq |L| + |E|$: return k.
5. return QuickSelect(L,$k - |L| - |E|$).

Quickselect

Problem: Find k-th smallest element of a sequence $(x_1, \ldots, x_n)$.



Sir Tony Horae

QuickSelect($(x_1, \ldots, x_n)$, k), Sir Tony Horae 1961

1. Choose pivot p.
2. Split $(x_1, \ldots, x_n)$ to $L = \{x_i : x_i < p\}$, $E = \{x_i : x_i = p\}$, $R = \{x_i : x_i > p\}$.
3. If $k \leq |L|$: return QuickSelect(L,k).
4. If $k \leq |L| + |E|$: return k.
5. return QuickSelect(L,$k - |L| - |E|$).

1. Assume that p is always maximum.

1. Assume that p is always maximum.
   Time complexity: $\Theta(N^2)$.

1. Assume that p is always maximum.
   Time complexity: $\Theta(N^2)$.
2. Assume that p is median.

1. Assume that p is always maximum.
   Time complexity: $\Theta(N^2)$.
2. Assume that p is median.
   Time complexity: $T(N) = T\left(\frac{N}{2}\right) + \Theta(N) = \Theta(N)$.

1. Assume that p is always maximum.
   Time complexity: $\Theta(N^2)$.
2. Assume that p is median.
   Time complexity: $T(N) = T\left(\frac{N}{2}\right) + \Theta(N) = \Theta(N)$.
3. Assume that p is "almost median" (at least $\frac{1}{4}$ of elements is smaller than p and $\frac{1}{4}$ is bigger than p).

1. Assume that p is always maximum.
   Time complexity: $\Theta(N^2)$.
2. Assume that p is median.
   Time complexity: $T(N) = T\left(\frac{N}{2}\right) + \Theta(N) = \Theta(N)$.
3. Assume that p is “almost median” (at least $\frac{1}{4}$ of elements is smaller than p and $\frac{1}{4}$ is bigger than p).
   Time complexity: $T(N) = T\left(\frac{3N}{2}\right) + \Theta(N) = \Theta(N)$.

1. Assume that p is always maximum.
   Time complexity: $\Theta(N^2)$.
2. Assume that p is median.
   Time complexity: $T(N) = T\left(\frac{N}{2}\right) + \Theta(N) = \Theta(N)$.
3. Assume that p is "almost median" (at least $\frac{1}{4}$ of elements is smaller than p and $\frac{1}{4}$ is bigger than p).
   Time complexity: $T(N) = T\left(\frac{3N}{2}\right) + \Theta(N) = \Theta(N)$.
4. Randomized choice of almost median: $\Pr[\text{random element is almost median}] \geq \frac{1}{2}$

1. Assume that p is always maximum.
   Time complexity: $\Theta(N^2)$.
2. Assume that p is median.
   Time complexity: $T(N) = T\left(\frac{N}{2}\right) + \Theta(N) = \Theta(N)$.
3. Assume that p is "almost median" (at least $\frac{1}{4}$ of elements is smaller than p and $\frac{1}{4}$ is bigger than p).
   Time complexity: $T(N) = T\left(\frac{3N}{2}\right) + \Theta(N) = \Theta(N)$.
4. Randomized choice of almost median: $\Pr[\text{random element is almost median}] \geq \frac{1}{2}$

Lemma

Let V be an event that occurs in trial with probability p. The expected number of trials to first occurrence of V is $\frac{1}{p}$.

Expected number of attempts; time complexity $\Theta(N)$.

1. Assume that p is always maximum.
   Time complexity: $\Theta(N^2)$.
2. Assume that p is median.
   Time complexity: $T(N) = T\left(\frac{N}{2}\right) + \Theta(N) = \Theta(N)$.
3. Assume that p is "almost median" (at least $\frac{1}{4}$ of elements is smaller than p and $\frac{1}{4}$ is bigger than p).
   Time complexity: $T(N) = T\left(\frac{3N}{2}\right) + \Theta(N) = \Theta(N)$.
4. Randomized choice of almost median: $\Pr[\text{random element is almost median}] \geq \frac{1}{2}$

Lemma

Let V be an event that occurs in trial with probability p. The expected number of trials to first occurrence of V is $\frac{1}{p}$.

Expected number of attempts; time complexity $\Theta(N)$.

Proof.

Method 1:
$\mathbb{E}[\#\text{ of trials}] = \sum_{i \geq 1} i \cdot \Pr[\text{we do precisely } i \text{ trials}]$

1. Assume that p is always maximum.
   Time complexity: $\Theta(N^2)$.
2. Assume that p is median.
   Time complexity: $T(N) = T\left(\frac{N}{2}\right) + \Theta(N) = \Theta(N)$.
3. Assume that p is "almost median" (at least $\frac{1}{4}$ of elements is smaller than p and $\frac{1}{4}$ is bigger than p).
   Time complexity: $T(N) = T\left(\frac{3N}{2}\right) + \Theta(N) = \Theta(N)$.
4. Randomized choice of almost median: Pr[random element is almost median] $\geq \frac{1}{2}$

Lemma

Let V be an event that occurs in trial with probability p. The expected number of trials to first occurrence of V is $\frac{1}{p}$.

Expected number of attempts; time complexity $\Theta(N)$.

Proof.

Method 1:

$\mathbb{E}[\#\text{ of trials}] = \sum_{i \geq 1} i \cdot \Pr[\text{we do precisely } i \text{ trials}]$

$\Pr[\text{we do precisely } i \text{ trials}] = (1-p)^i p$

1. Assume that p is always maximum.
   Time complexity: $\Theta(N^2)$.
2. Assume that p is median.
   Time complexity: $T(N) = T\left(\frac{N}{2}\right) + \Theta(N) = \Theta(N)$.
3. Assume that p is "almost median" (at least $\frac{1}{4}$ of elements is smaller than p and $\frac{1}{4}$ is bigger than p).
   Time complexity: $T(N) = T\left(\frac{3N}{2}\right) + \Theta(N) = \Theta(N)$.
4. Randomized choice of almost median: $\Pr[\text{random element is almost median}] \geq \frac{1}{2}$

Lemma

Let V be an event that occurs in trial with probability p. The expected number of trials to first occurrence of V is $\frac{1}{p}$.

Expected number of attempts; time complexity $\Theta(N)$.

Proof.

Method 1:
$\mathbb{E}[\#\text{ of trials}] = \sum_{i\geq 1} i \cdot \Pr[\text{we do precisely } i \text{ trials}]$
$\Pr[\text{we do precisely } i \text{ trials}] = (1-p)^i p$
Method 2:
$\mathbb{E}[\#\text{ of trials}] = D = 1 + p0 + (1-p)D$

1. Assume that p is always maximum.
   Time complexity: $\Theta(N^2)$.
2. Assume that p is median.
   Time complexity: $T(N) = T\left(\frac{N}{2}\right) + \Theta(N) = \Theta(N)$.
3. Assume that p is "almost median" (at least $\frac{1}{4}$ of elements is smaller than p and $\frac{1}{4}$ is bigger than p).
   Time complexity: $T(N) = T\left(\frac{3N}{2}\right) + \Theta(N) = \Theta(N)$.
4. Randomized choice of almost median: $\Pr[\text{random element is almost median}] \geq \frac{1}{2}$

Lemma

Let V be an event that occurs in trial with probability p. The expected number of trials to first occurrence of V is $\frac{1}{p}$.

Expected number of attempts; time complexity $\Theta(N)$.

Proof.

Method 1:
$\mathbb{E}[\# \text{ of trials}] = \sum_{i \geq 1} i \cdot \Pr[\text{we do precisely } i \text{ trials}]$
$\Pr[\text{we do precisely } i \text{ trials}] = (1-p)^i p$
Method 2:
$\mathbb{E}[\# \text{ of trials}] = D = 1 + p0 + (1-p)D$
$D(1 + 1 + q) = 1$

1. Assume that p is always maximum.
   Time complexity: $\Theta(N^2)$.
2. Assume that p is median.
   Time complexity: $T(N) = T\left(\frac{N}{2}\right) + \Theta(N) = \Theta(N)$.
3. Assume that p is "almost median" (at least $\frac{1}{4}$ of elements is smaller than p and $\frac{1}{4}$ is bigger than p).
   Time complexity: $T(N) = T\left(\frac{3N}{2}\right) + \Theta(N) = \Theta(N)$.
4. Randomized choice of almost median: $\Pr[\text{random element is almost median}] \geq \frac{1}{2}$

Lemma

Let V be an event that occurs in trial with probability p. The expected number of trials to first occurrence of V is $\frac{1}{p}$.

Expected number of attempts; time complexity $\Theta(N)$.

Proof.

Method 1:
$\mathbb{E}[\#\text{ of trials}] = \sum_{i\geq 1} i \cdot \Pr[\text{we do precisely } i \text{ trials}]$
$\Pr[\text{we do precisely } i \text{ trials}] = (1-p)^i p$
Method 2:
$\mathbb{E}[\#\text{ of trials}] = D = 1 + p0 + (1-p)D$
$D(1+1+q) = 1$
$D = \frac{1}{q}$ □

1. Assume that p is always maximum.
   Time complexity: $\Theta(N^2)$.
2. Assume that p is median.
   Time complexity: $T(N) = T\left(\frac{N}{2}\right) + \Theta(N) = \Theta(N)$.
3. Assume that p is "almost median" (at least $\frac{1}{4}$ of elements is smaller than p and $\frac{1}{4}$ is bigger than p).
   Time complexity: $T(N) = T\left(\frac{3N}{2}\right) + \Theta(N) = \Theta(N)$.
4. Randomized choice of almost median: $\Pr[\text{random element is almost median}] \geq \frac{1}{2}$

Lemma

Let V be an event that occurs in trial with probability p. The expected number of trials to first occurrence of V is $\frac{1}{p}$.

   Expected number of attempts; time complexity $\Theta(N)$.
5. Random choice of pivot.
   Stage of algorithm ends by finding almost median.

1. Assume that p is always maximum.
   Time complexity: $\Theta(N^2)$.
2. Assume that p is median.
   Time complexity: $T(N) = T\left(\frac{N}{2}\right) + \Theta(N) = \Theta(N)$.
3. Assume that p is "almost median" (at least $\frac{1}{4}$ of elements is smaller than p and $\frac{1}{4}$ is bigger than p).
   Time complexity: $T(N) = T\left(\frac{3N}{2}\right) + \Theta(N) = \Theta(N)$.
4. Randomized choice of almost median: $\Pr[\text{random element is almost median}] \geq \frac{1}{2}$

Lemma

Let V be an event that occurs in trial with probability p. The expected number of trials to first occurrence of V is $\frac{1}{p}$.

   Expected number of attempts; time complexity $\Theta(N)$.
5. Random choice of pivot.
   Stage of algorithm ends by finding almost median.

1. Assume that p is always maximum.
   Time complexity: $\Theta(N^2)$.
2. Assume that p is median.
   Time complexity: $T(N) = T\left(\frac{N}{2}\right) + \Theta(N) = \Theta(N)$.
3. Assume that p is "almost median" (at least $\frac{1}{4}$ of elements is smaller than p and $\frac{1}{4}$ is bigger than p).
   Time complexity: $T(N) = T\left(\frac{3N}{2}\right) + \Theta(N) = \Theta(N)$.
4. Randomized choice of almost median: Pr[random element is almost median] $\geq \frac{1}{2}$

Lemma

Let V be an event that occurs in trial with probability p. The expected number of trials to first occurrence of V is $\frac{1}{p}$.

   Expected number of attempts; time complexity $\Theta(N)$.
5. Random choice of pivot.
   Stage of algorithm ends by finding almost median.
   $\mathbb{E}[\#\text{ steps in stage}] = 2$

1. Assume that p is always maximum.
   Time complexity: $\Theta(N^2)$.
2. Assume that p is median.
   Time complexity: $T(N) = T\left(\frac{N}{2}\right) + \Theta(N) = \Theta(N)$.
3. Assume that p is "almost median" (at least $\frac{1}{4}$ of elements is smaller than p and $\frac{1}{4}$ is bigger than p).
   Time complexity: $T(N) = T\left(\frac{3N}{2}\right) + \Theta(N) = \Theta(N)$.
4. Randomized choice of almost median: Pr[random element is almost median] $\geq \frac{1}{2}$

Lemma

Let V be an event that occurs in trial with probability p. The expected number of trials to first occurrence of V is $\frac{1}{p}$.

   Expected number of attempts; time complexity $\Theta(N)$.
5. Random choice of pivot.
   Stage of algorithm ends by finding almost median.
   $\mathbb{E}[\#\text{ steps in stage}] = 2$
   $\mathbb{E}[\text{time for stage}] = \Theta(N)$

1. Assume that p is always maximum.
   Time complexity: $\Theta(N^2)$.
2. Assume that p is median.
   Time complexity: $T(N) = T\left(\frac{N}{2}\right) + \Theta(N) = \Theta(N)$.
3. Assume that p is "almost median" (at least $\frac{1}{4}$ of elements is smaller than p and $\frac{1}{4}$ is bigger than p).
   Time complexity: $T(N) = T\left(\frac{3N}{2}\right) + \Theta(N) = \Theta(N)$.
4. Randomized choice of almost median: $\Pr[\text{random element is almost median}] \geq \frac{1}{2}$

Lemma

Let V be an event that occurs in trial with probability p. The expected number of trials to first occurrence of V is $\frac{1}{p}$.

   Expected number of attempts; time complexity $\Theta(N)$.
5. Random choice of pivot.
   Stage of algorithm ends by finding almost median.
   $\mathbb{E}[\#\text{ steps in stage}] = 2$
   $\mathbb{E}[\text{time for stage}] = \Theta(N)$
   Every stage reduces problem to $\frac{3}{4}$.
   Time complexity: $\Theta(N)$.

Median of medians algorithm

Select($(x_1, \ldots, x_n), k$), Blum, Floyd, Pratt, Rivest, Tarjan 1973

1. If $n \leq 10$ use brute-force.
2. Divide $x_1, \ldots x_n$ to 5-tuples $P_1, \ldots P_t$, $t = \lfloor \frac{n}{b} \rfloor$.
3. For $i = 1, 2, \ldots t$: $m_i \leftarrow$Median(P_i).
4. $p \leftarrow$Select($(m_1, \ldots, m_t)$, $\left\lfloor \frac{t}{2} \right\rfloor$)

Median of medians algorithm

Select($(x_1, \ldots, x_n)$, k), Blum, Floyd, Pratt, Rivest, Tarjan 1973

1. If $n \leq 10$ use brute-force.
2. Divide $x_1, \ldots x_n$ to 5-tuples $P_1, \ldots P_t$, $t = \left\lfloor \frac{n}{b} \right\rfloor$.
3. For $i = 1, 2, \ldots t$: $m_i \leftarrow$Median(P_i).
4. $p \leftarrow$Select($(m_1, \ldots, m_t)$, $\left\lfloor \frac{t}{2} \right\rfloor$)
5. Split $(x_1, \ldots, x_n)$ to $L = \{x_i : x_i < p\}$, $E = \{x_i : x_i = p\}$, $R = \{x_i : x_i > p\}$.
6. If $k \leq |L|$: return Select(L,k).
7. If $k \leq |L| + |E|$: return k.
8. return Select(L,$k - |L| - |E|$).

Median of medians algorithm

$$T(n) = T(\frac{n}{5}) + T(\frac{7}{10}n) + \Theta(n)$$

Median of medians algorithm

$$T(n) = T(\frac{n}{5}) + T(\frac{7}{10}n) + \Theta(n)$$

Assume $T(n) = cn$

$$cn = \frac{1}{5}cn + \frac{7}{10}cn + \alpha n$$

Median of medians algorithm

$$T(n) = T(\frac{n}{5}) + T(\frac{7}{10}n) + \Theta(n)$$

Assume $T(n) = cn$

$$\begin{aligned} cn &= \frac{1}{5}cn + \frac{7}{10}cn + \alpha n \\ \left(1 - \frac{1}{5} - \frac{7}{10}\right) cn &= \alpha n \end{aligned}$$

Median of medians algorithm

$$T(n) = T(\frac{n}{5}) + T(\frac{7}{10}n) + \Theta(n)$$

Assume $T(n) = cn$

$$\begin{aligned} cn &= \frac{1}{5}cn + \frac{7}{10}cn + \alpha n \\ \left(1 - \frac{1}{5} - \frac{7}{10}\right)cn &= \alpha n \\ \frac{1}{10}cn &= \alpha n \end{aligned}$$

Median of medians algorithm

$$T(n) = T(\frac{n}{5}) + T(\frac{7}{10}n) + \Theta(n)$$

Assume $T(n) = cn$

$$\begin{aligned} cn &= \frac{1}{5}cn + \frac{7}{10}cn + \alpha n \\ \left(1 - \frac{1}{5} - \frac{7}{10}\right) cn &= \alpha n \\ \frac{1}{10} cn &= \alpha n \\ c = 10\alpha \end{aligned}$$

Median of medians algorithm

$$T(n) = T(\frac{n}{5}) + T(\frac{7}{10}n) + \Theta(n)$$

Assume $T(n) = cn$

$$\begin{aligned} cn &= \frac{1}{5}cn + \frac{7}{10}cn + \alpha n \\ \left(1 - \frac{1}{5} - \frac{7}{10}\right) cn &= \alpha n \\ \frac{1}{10} cn &= \alpha n \\ c = 10\alpha \end{aligned}$$

Deterministic runtime: $\Theta(n)$!