CS3383 Lecture 1.4: Order Statistics

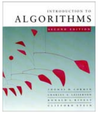
David Bremner

January 28, 2024

Outline

Even More Divide and Conquer

- Randomized median finding
- Median of medians



Order statistics

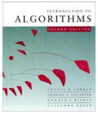
Select the ith smallest of n elements (the element with ***rank*** i).

- $i = 1$: ***minimum***;
- $i = n$: ***maximum***;
- $i = \lfloor (n+1)/2 \rfloor$ or $\lceil (n+1)/2 \rceil$: ***median***.

Naive algorithm: Sort and index ith element.

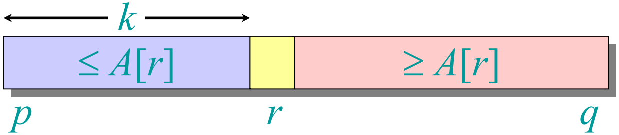
Worst-case running time $= \Theta(n \lg n) + \Theta(1)$
$= \Theta(n \lg n)$,

using merge sort or heapsort (*not* quicksort).



Randomized divide-and-conquer algorithm

```
RAND-SELECT($A, p, q, i$)        ▷ $i$th smallest of $A[p\,..\,q]$
    if $p = q$ then return $A[p]$
    $r \leftarrow$ RAND-PARTITION($A, p, q$)
    $k \leftarrow r - p + 1$               ▷ $k = \text{rank}(A[r])$
    if $i = k$ then return $A[r]$
    if $i < k$
        then return RAND-SELECT($A, p, r - 1, i$)
        else return RAND-SELECT($A, r + 1, q, i - k$)
```



Example

Select the $i = 7$th smallest:

6	10	13	5	8	3	2	11

$i = 7$

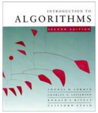
pivot

Partition:

2	5	3	6	8	13	10	11

$k = 4$

Select the $7 - 4 = 3$rd smallest recursively.



Intuition for analysis

(All our analyses today assume that all elements are distinct.)

Lucky:

$$T(n) = T(9n/10) + \Theta(n) \qquad n^{\log_{10/9} 1} = n^0 = 1$$

$$= \Theta(n) \qquad \text{Case 3}$$

Unlucky:

$$T(n) = T(n-1) + \Theta(n) \qquad \text{arithmetic series}$$

$$= \Theta(n^2)$$

Worse than sorting!

Randomized median finding

```python
def select(A,p,q,i):
    n = q - p + 1;  bad = True
    if n==1: return A[p]
    while bad:
        r = partition(A,p,q,randrange(p,q))
        k = r - p
        if (k == i): return A[r]
        bad = (k< n//4) or (k> 3*n//4)
    if (i < k):
        return select(A,p,r-1,i)
    else:
        return select(A,r+1,q,i-k-1)
```

How many times do we partition?

```
def select(A,p,q,i):
    ⋮
    while bad:
        ⋮
        bad = (k< n//4) or (k> 3*n//4)
    if (i < k):
        return select(A,p,r-1,i)
    else:
        return select(A,r+1,q,i-k-1)
```

- Call a pivot r good if $\lfloor n/4 \rfloor$ elements are on either side.
- Odds are $50/50$.

A random recurrence

- Let $W(n)$ be the random variable for time in `while`.
- Let $s = 4/3$

$$
\begin{aligned}
T(n) &\le W(n) + T(n/s) + O(1) \\
&\le W(n) + W(n/s) + T(n/s^2) + O(1) + O(1) \\
&\le \sum_{j=0}^{\log_s(n)} \left[W(\frac{n}{s^j}) + O(1) \right] \\
&\le \left[\sum_{j=0}^{\log_s(n)} W\left(\frac{n}{s^j}\right) \right] + c_1 \log_s n \quad \forall n \ge n_0
\end{aligned}
$$

A random recurrence

- Let $W(n)$ be the random variable for time in `while`.
- Let $s = 4/3$

$$
\begin{aligned}
T(n) &\le W(n) + T(n/s) + O(1) \\
&\le W(n) + W(n/s) + T(n/s^2) + O(1) + O(1) \\
&\le \sum_{j=0}^{\log_s(n)} \left[W(\frac{n}{s^j}) + O(1) \right] \\
&\le \left[\sum_{j=0}^{\log_s(n)} W\left(\frac{n}{s^j}\right) \right] + c_1 \log_s n \quad \forall n \ge n_0
\end{aligned}
$$

A random recurrence

- Let $W(n)$ be the random variable for time in `while`.
- Let $s = 4/3$

$$
\begin{aligned}
T(n) &\leq W(n) + T(n/s) + O(1) \\
&\leq W(n) + W(n/s) + T(n/s^2) + O(1) + O(1) \\
&\leq \sum_{j=0}^{\log_s(n)} \left[W(\frac{n}{s^j}) + O(1) \right] \\
&\leq \left[\sum_{j=0}^{\log_s(n)} W\left(\frac{n}{s^j}\right) \right] + c_1 \log_s n \quad \forall n \geq n_0
\end{aligned}
$$

A random recurrence

- Let $W(n)$ be the random variable for time in `while`.
- Let $s = 4/3$

$$
\begin{aligned}
T(n) &\leq W(n) + T(n/s) + O(1) \\
&\leq W(n) + W(n/s) + T(n/s^2) + O(1) + O(1) \\
&\leq \sum_{j=0}^{\log_s(n)} \left[W(\frac{n}{s^j}) + O(1) \right] \\
&\leq \left[\sum_{j=0}^{\log_s(n)} W\left(\frac{n}{s^j}\right) \right] + c_1 \log_s n \quad \forall n \geq n_0
\end{aligned}
$$

Linearity of expectation, again

- Let $W(n)$ be the random variable for time in `while`.
- Let $s = 4/3$

For all $n \geq n_0$

$$T(n) \leq \left(\sum_{j=0}^{\log_s(n)} W\left(\frac{n}{s^j}\right) \right) + c_1 \log_s n$$

$$E[T(n)] \leq E\left[\sum_{j=0}^{\log_s(n)} W\left(\frac{n}{s^j}\right) \right] + c_1 \log_s n$$

$$\leq \sum_{j=0}^{\log_s(n)} E\left[W\left(\frac{n}{s^j}\right) \right] + c_1 \log_s n$$

Linearity of expectation, again

- Let $W(n)$ be the random variable for time in `while`.
- Let $s = 4/3$

For all $n \geq n_0$

$$T(n) \leq \left(\sum_{j=0}^{\log_s(n)} W\left(\frac{n}{s^j}\right) \right) + c_1 \log_s n$$

$$E[T(n)] \leq E\left[\sum_{j=0}^{\log_s(n)} W\left(\frac{n}{s^j}\right) \right] + c_1 \log_s n$$

$$\leq \sum_{j=0}^{\log_s(n)} E\left[W\left(\frac{n}{s^j}\right) \right] + c_1 \log_s n$$

Linearity of expectation, again

- Let $W(n)$ be the random variable for time in `while`.
- Let $s = 4/3$

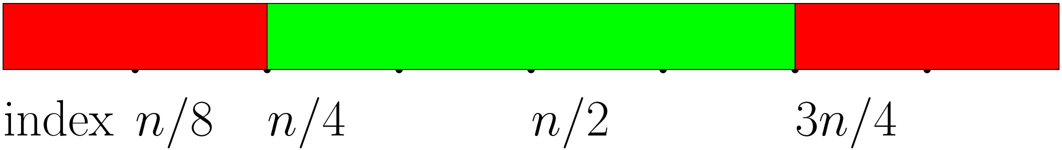
For all $n \geq n_0$

$$T(n) \leq \left(\sum_{j=0}^{\log_s(n)} W\left(\frac{n}{s^j}\right) \right) + c_1 \log_s n$$

$$E[T(n)] \leq E\left[\sum_{j=0}^{\log_s(n)} W\left(\frac{n}{s^j}\right) \right] + c_1 \log_s n$$

$$\leq \sum_{j=0}^{\log_s(n)} E\left[W\left(\frac{n}{s^j}\right)\right] + c_1 \log_s n$$

How many iterations?

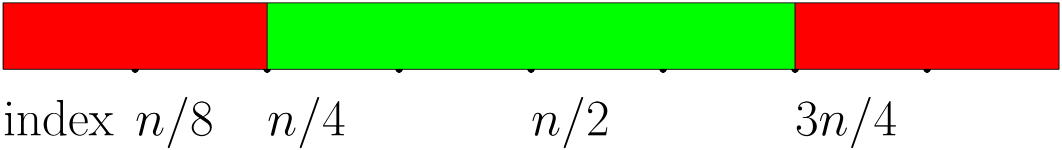


$$W(n) \le c_2 n \cdot \sum_{j=1}^{\infty} j X_j$$

where

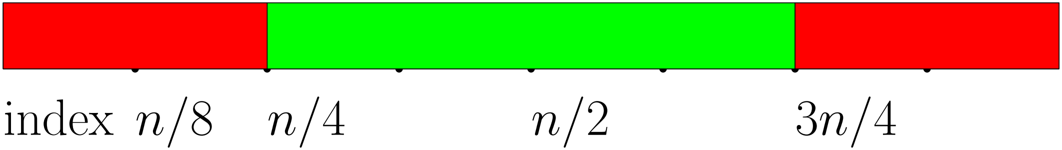
$$X_j = \begin{cases} 1 & \texttt{while} \text{ runs } j \text{ times} \\ 0 & \text{otherwise} \end{cases}$$

How many expected iterations?



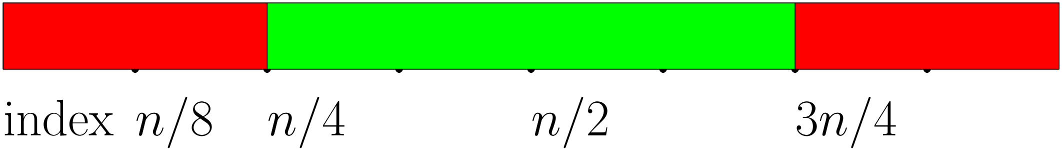
$$\begin{aligned} E[W(n)] &\leq E\left[c_2 n \cdot \sum_{j=1}^{\infty} j X_j\right] \\ &\leq c_2 n \cdot \sum_{j=1}^{\infty} E[j X_j] \\ &\leq c_2 n \cdot \sum_{j=1}^{\infty} j P[X_j = 1] \end{aligned}$$

How many expected iterations?



$$\begin{aligned} E[W(n)] &\leq E\left[c_2 n \cdot \sum_{j=1}^{\infty} jX_j\right] \\ &\leq c_2 n \cdot \sum_{j=1}^{\infty} E[jX_j] \\ &\leq c_2 n \cdot \sum_{j=1}^{\infty} jP[X_j = 1] \end{aligned}$$

How many expected iterations?

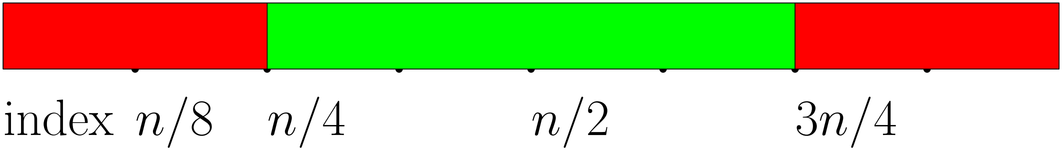


$$E[W(n)] \leq E\left[c_2 n \cdot \sum_{j=1}^{\infty} jX_j\right]$$

$$\leq c_2 n \cdot \sum_{j=1}^{\infty} E[jX_j]$$

$$\leq c_2 n \cdot \sum_{j=1}^{\infty} jP[X_j = 1]$$

Probability of exactly j iterations



$$X_j = \begin{cases} 1 & \texttt{while} \text{ runs } j \text{ times} \\ 0 & \text{otherwise} \end{cases}$$

$$\begin{aligned} P[X_j = 1] &= (1-p)^{j-1} \cdot p \\ &= \frac{1}{2^j} \end{aligned}$$

How many expected iterations? (redux)

$$\begin{aligned}
E[W(n)] &\leq c_2 n \cdot \sum_{j=1}^{\infty} j P[X_j = 1] \\
&\leq c_2 n \cdot \sum_{j=1}^{\infty} \frac{j}{2^j} \\
&\leq c_2 n \cdot \frac{1/2}{(1-1/2)^2} \\
&\leq c_3 n
\end{aligned}$$

(CLRS4 A.11)

How many expected iterations? (redux)

$$
\begin{aligned}
E[W(n)] &\leq c_2 n \cdot \sum_{j=1}^{\infty} j P[X_j = 1] \\
&\leq c_2 n \cdot \sum_{j=1}^{\infty} \frac{j}{2^j} \\
&\leq c_2 n \cdot \frac{1/2}{(1-1/2)^2} \\
&\leq c_3 n
\end{aligned}
$$

(CLRS4 A.11)

How many expected iterations? (redux)

$$\begin{aligned} E[W(n)] &\le c_2 n \cdot \sum_{j=1}^{\infty} j P[X_j = 1] \\ &\le c_2 n \cdot \sum_{j=1}^{\infty} \frac{j}{2^j} \\ &\le c_2 n \cdot \frac{1/2}{(1-1/2)^2} \\ &\le c_3 n \end{aligned}$$

(CLRS4 A.11)

How many expected iterations? (redux)

$$
\begin{aligned}
E[W(n)] &\leq c_2 n \cdot \sum_{j=1}^{\infty} j P[X_j = 1] \\
&\leq c_2 n \cdot \sum_{j=1}^{\infty} \frac{j}{2^j} \\
&\leq c_2 n \cdot \frac{1/2}{(1-1/2)^2} \\
&\leq c_3 n
\end{aligned}
$$

(CLRS4 A.11)

Geometric series, again

Let $W(n)$ be time in `while`. Let $s = 4/3$. For all $n \geq n_0$:

$$E[T(n)] \leq \sum_{j=0}^{\log_s(n)} E\left[W\left(\frac{n}{s^j}\right)\right] + c_1 \log_s n$$

$$\leq \left(\sum_{j=0}^{\log_s(n)} c_3 \frac{n}{s^j}\right) + c_1 \log_s n$$

(CLRS4-A.7) $$\leq c_3 n \cdot \frac{1}{1 - s^{-1}} + c_1 \log_s n$$

Geometric series, again

Let $W(n)$ be time in `while`. Let $s = 4/3$. For all $n \geq n_0$:

$$E[T(n)] \leq \sum_{j=0}^{\log_s(n)} E\left[W\left(\frac{n}{s^j}\right)\right] + c_1 \log_s n$$

$$\leq \left(\sum_{j=0}^{\log_s(n)} c_3 \frac{n}{s^j}\right) + c_1 \log_s n$$

(CLRS4-A.7) $$\leq c_3 n \cdot \frac{1}{1 - s^{-1}} + c_1 \log_s n$$

Geometric series, again

Let $W(n)$ be time in `while`. Let $s = 4/3$. For all $n \geq n_0$:

$$E[T(n)] \leq \sum_{j=0}^{\log_s(n)} E\left[W\left(\frac{n}{s^j}\right)\right] + c_1 \log_s n$$

$$\leq \left(\sum_{j=0}^{\log_s(n)} c_3 \frac{n}{s^j}\right) + c_1 \log_s n$$

(CLRS4-A.7) $$\leq c_3 n \cdot \frac{1}{1 - s^{-1}} + c_1 \log_s n$$

Deterministically choosing a good pivot.

- it turns out we can achieve $O(n)$ time deterministically

Deterministically choosing a good pivot.

- it turns out we can achieve $O(n)$ time deterministically
- deterministic algorithm is more complicated

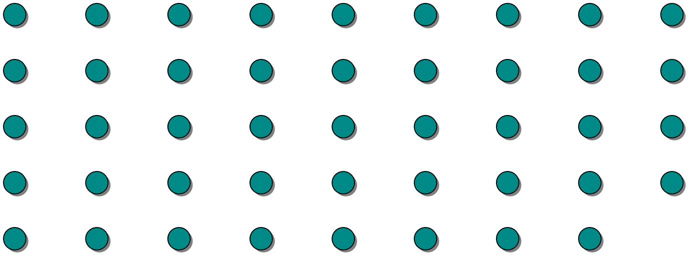
Deterministically choosing a good pivot.

- it turns out we can achieve $O(n)$ time deterministically
- deterministic algorithm is more complicated
- practical performance is typically worse

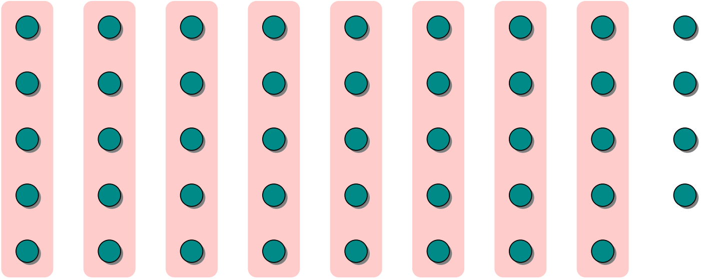
Deterministically choosing a good pivot.

- it turns out we can achieve $O(n)$ time deterministically
- deterministic algorithm is more complicated
- practical performance is typically worse
- The main idea of this algorithm is taking the median of medians

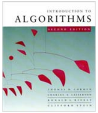
Choosing the pivot



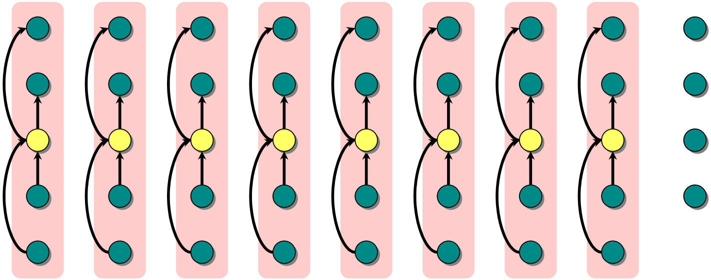
Choosing the pivot



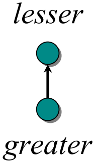
1. Divide the n elements into groups of 5.

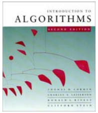


Choosing the pivot

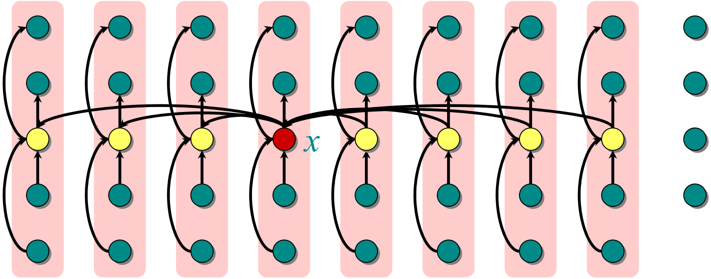


1. Divide the n elements into groups of 5. Find the median of each 5-element group by rote.

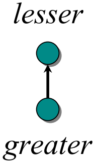


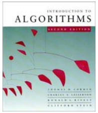


Choosing the pivot

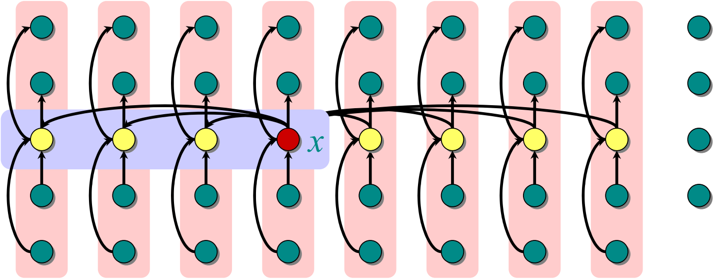


1. Divide the n elements into groups of 5. Find the median of each 5-element group by rote.
2. Recursively SELECT the median x of the $\lfloor n/5 \rfloor$ group medians to be the pivot.

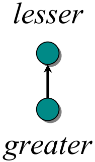


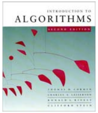


Analysis

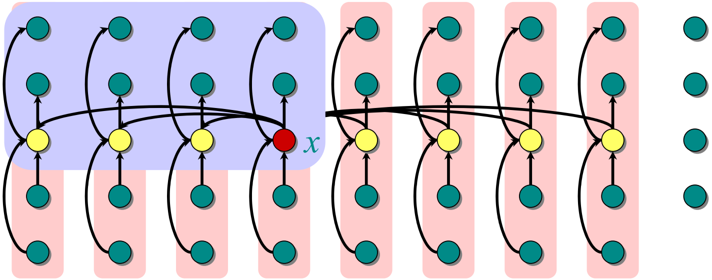


At least half the group medians are $\leq x$, which is at least $\lfloor \lfloor n/5 \rfloor /2 \rfloor = \lfloor n/10 \rfloor$ group medians.



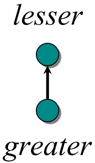


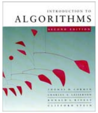
Analysis (Assume all elements are distinct.)



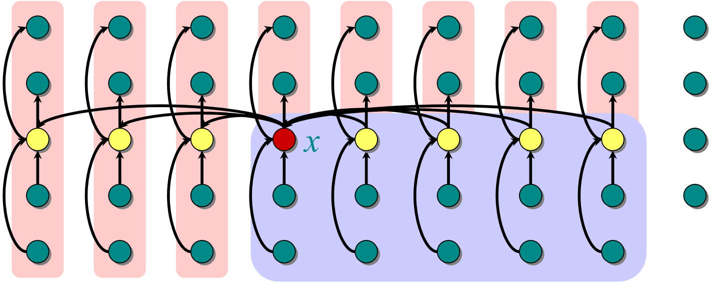
At least half the group medians are $\leq x$, which is at least $\lfloor \lfloor n/5 \rfloor /2 \rfloor = \lfloor n/10 \rfloor$ group medians.

- Therefore, at least $3\lfloor n/10 \rfloor$ elements are $\leq x$.



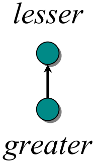


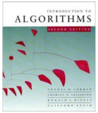
Analysis (Assume all elements are distinct.)



At least half the group medians are $\leq x$, which is at least $\lfloor \lfloor n/5 \rfloor /2 \rfloor = \lfloor n/10 \rfloor$ group medians.

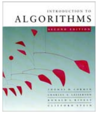
- Therefore, at least $3\lfloor n/10 \rfloor$ elements are $\leq x$.
- Similarly, at least $3\lfloor n/10 \rfloor$ elements are $\geq x$.





Minor simplification

- For $n \geq 50$, we have $3\lfloor n/10 \rfloor \geq n/4$.
- Therefore, for $n \geq 50$ the recursive call to SELECT in Step 4 is executed recursively on $\leq 3n/4$ elements.
- Thus, the recurrence for running time can assume that Step 4 takes time $T(3n/4)$ in the worst case.
- For $n < 50$, we know that the worst-case time is $T(n) = \Theta(1)$.



Developing the recurrence

```
$T(n)$        SELECT$(i, n)$
$\Theta(n)$      1. Divide the $n$ elements into groups of $5$.  Find
                 the median of each $5$-element group by rote.
$T(n/5)$      2. Recursively SELECT the median $x$ of the $\lfloor n/5 \rfloor$
                 group medians to be the pivot.
$\Theta(n)$      3. Partition around the pivot $x$.  Let $k = \text{rank}(x)$.
              4. if $i = k$ then return $x$
                 elseif $i < k$
$T(3n/4)$                then recursively SELECT the $i$th
                              smallest element in the lower part
                         else recursively SELECT the $(i–k)$th
                              smallest element in the upper part
```

A familiar recurrence

$$\begin{array}{lrl} & T(n) &= T(n/5) + T(3n/4) + cn \\ \text{(Guess)} & T(n) &\leq dn, n \geq n_0 \\ \text{(Strong induction)} & &\leq (1/5)dn + (3/4)dn + cn \\ \text{(Choice of } c\text{, } d\text{)} & &\leq dn \end{array}$$