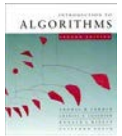
CS 381 – FALL 2019

Week 1.2, Wed, Aug 23

Related Notes on Linear Time Selection

Guest Lecture by Prof. Atallah

- Prof. Atallah covered linear time selection, but did not use power point in his lecture
- Several of you have requested slides/notes
- The slides in this deck are adapted from an earlier semester
    - Adapted from Erik Demaine, Charles Leiserson



Worst-case linear-time order statistics: Compute the i’th largest element of the array

Eg., the (n/2)‘th element, the (log n)’th element, etc.

SELECT(i, n)

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 $n/5$ group medians to be the pivot.
3. Partition around the pivot x. Let k = rank(x).
4. **if** $i = k$ **then return** x

   elseif $i < k$
   then recursively SELECT the ith smallest element in the lower part
   else recursively SELECT the $(i–k)$th smallest element in the upper part

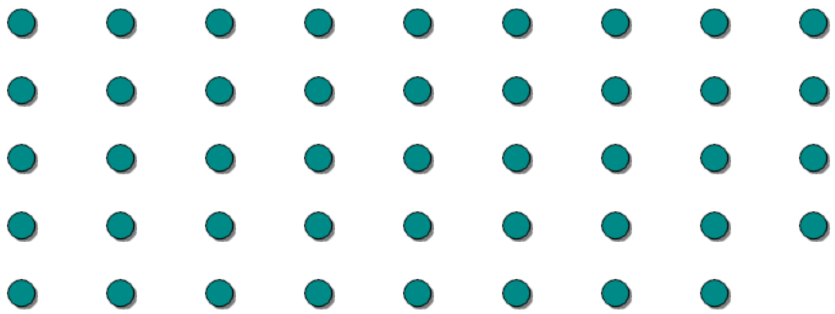
Note: The algorithm is due to Blum, Floyd, Pratt, Rivest, and Tarjan (1973).

Example

- SELECT(12, 15, A)
- A=[1 5 2 3 4, 21 20 22 23 24, 12 13 17 18 19]
- A=[1 5 2 3 4, 21 20 22 23 24, 12 13 18 17 19] ←medians
- A=[1 5 2 3 4, 21 20 22 23 24, 12 13 18 **17** 19] ←MoM
- Rank(17)=8
- 8 < 12 => SELECT(12-8, 15-8, [21 20 22 23 24 18 19])
- …→21.

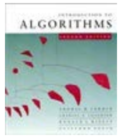
Analysis

Choosing the pivot

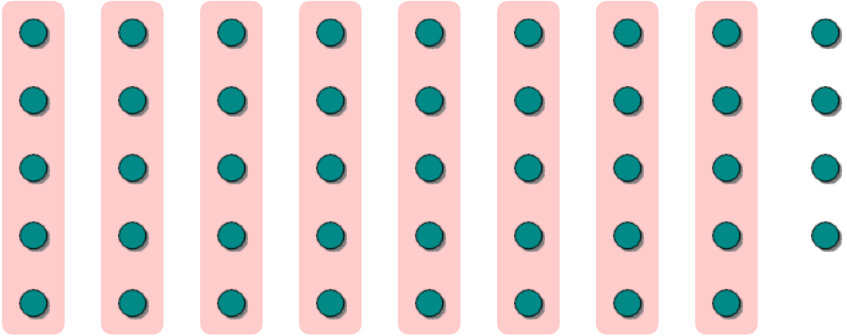


1. Divide the n elements into groups of 5.

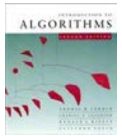
$\Theta(n)$



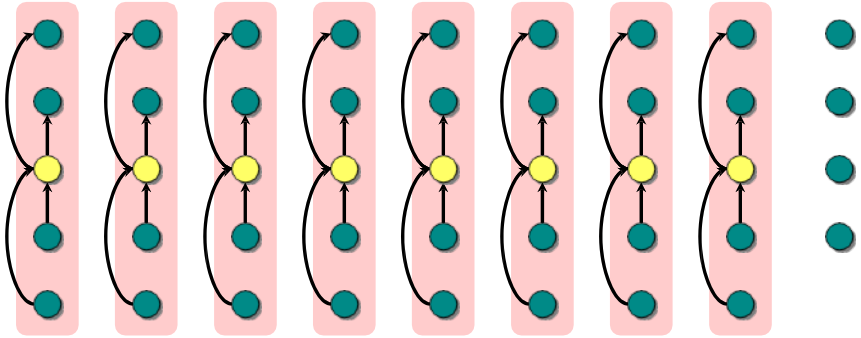
Choosing the pivot



1. Divide the n elements into groups of 5.



Choosing the pivot

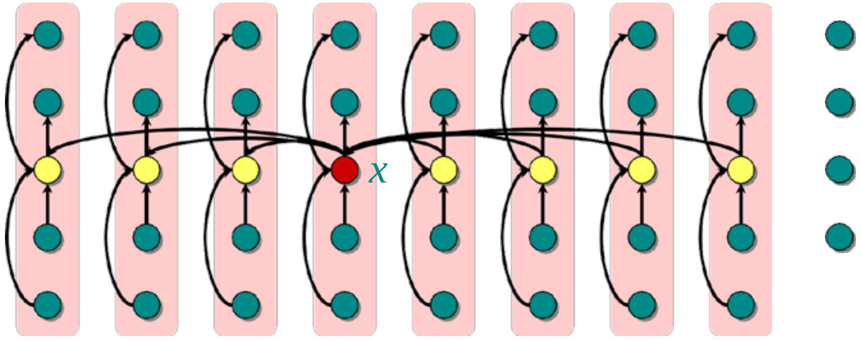


1. Divide the n elements into groups of 5.

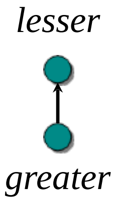
   Find the median of each 5-element group by brute force (rote).

$\Theta(n)$

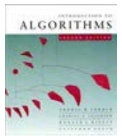
Choosing the pivot



First recursion:
finding a median of n/5 elements



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 n/5 group medians to be the pivot. (BIG IDEA)



Worst-case linear-time order statistics: Compute the i'th largest element of the array

Eg., the (n/2)'th element, the (log n)'th element, etc.

SELECT(i, n)

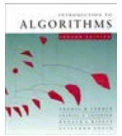
1. Divide the n elements into groups of 5. Find the median of each 5-element group by rote.
2. Recursively SELECT the median x of the n/5 group medians to be the pivot.
3. Partition around the pivot x. Let k = rank(x).
4. **if** $i = k$ **then return** x

   elseif $i < k$
   then recursively SELECT the ith smallest element in the lower part (indices up to k)
   else recursively SELECT the $(i–k)$th smallest element in the upper part (indices above k)

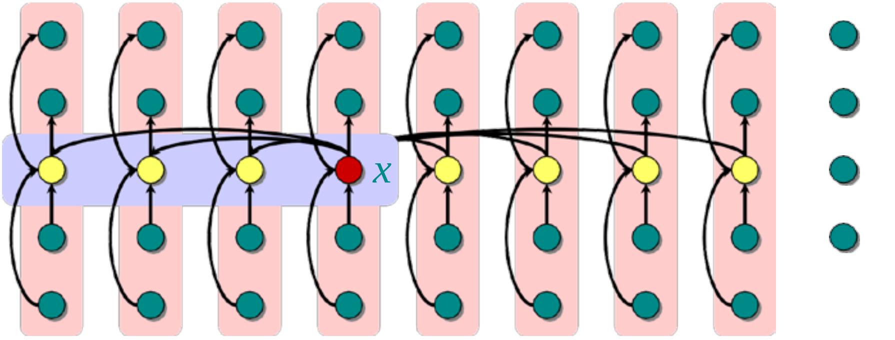
SELECT(t/2, t), where t=$\lfloor n/5 \rfloor$

Second recursion: on a smaller array

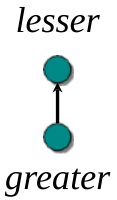
Analysis



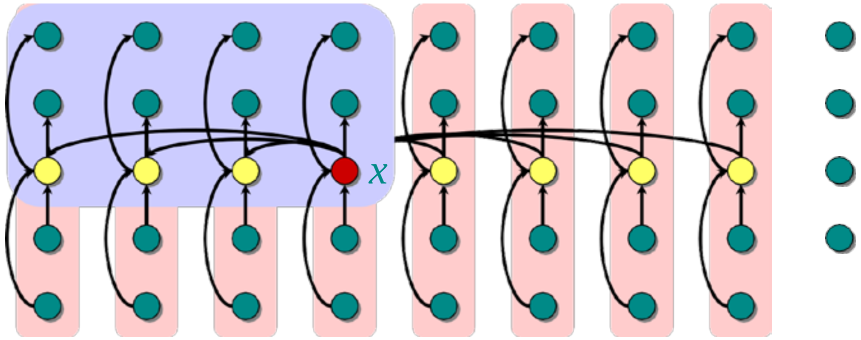
Question:
How balanced is the resulting array
if we run Partition with
median of medians (i.e., x) as the pivot?



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

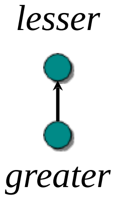


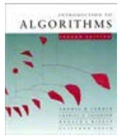
Analysis (Assume all elements are distinct.)



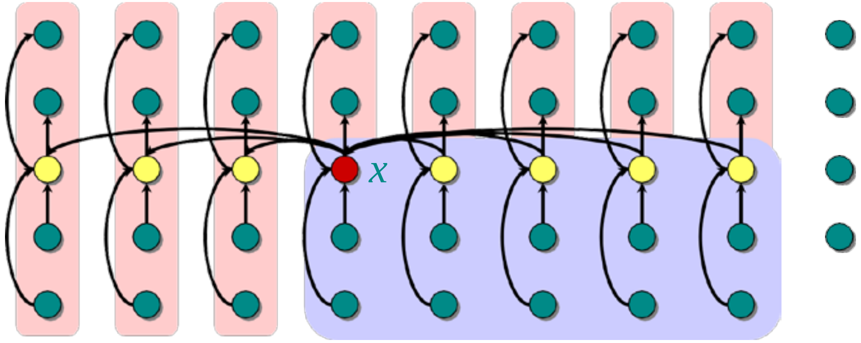
At least half the group medians are $\leq x$, which is at least $\lfloor \lfloor n/5 \rfloor /2 \rfloor = \lfloor n/10 \rfloor$ medians

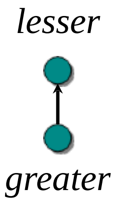
- Therefore, at least $3\lfloor n/10 \rfloor$ elements are $\leq x$.





Analysis (Assume all elements are distinct.)

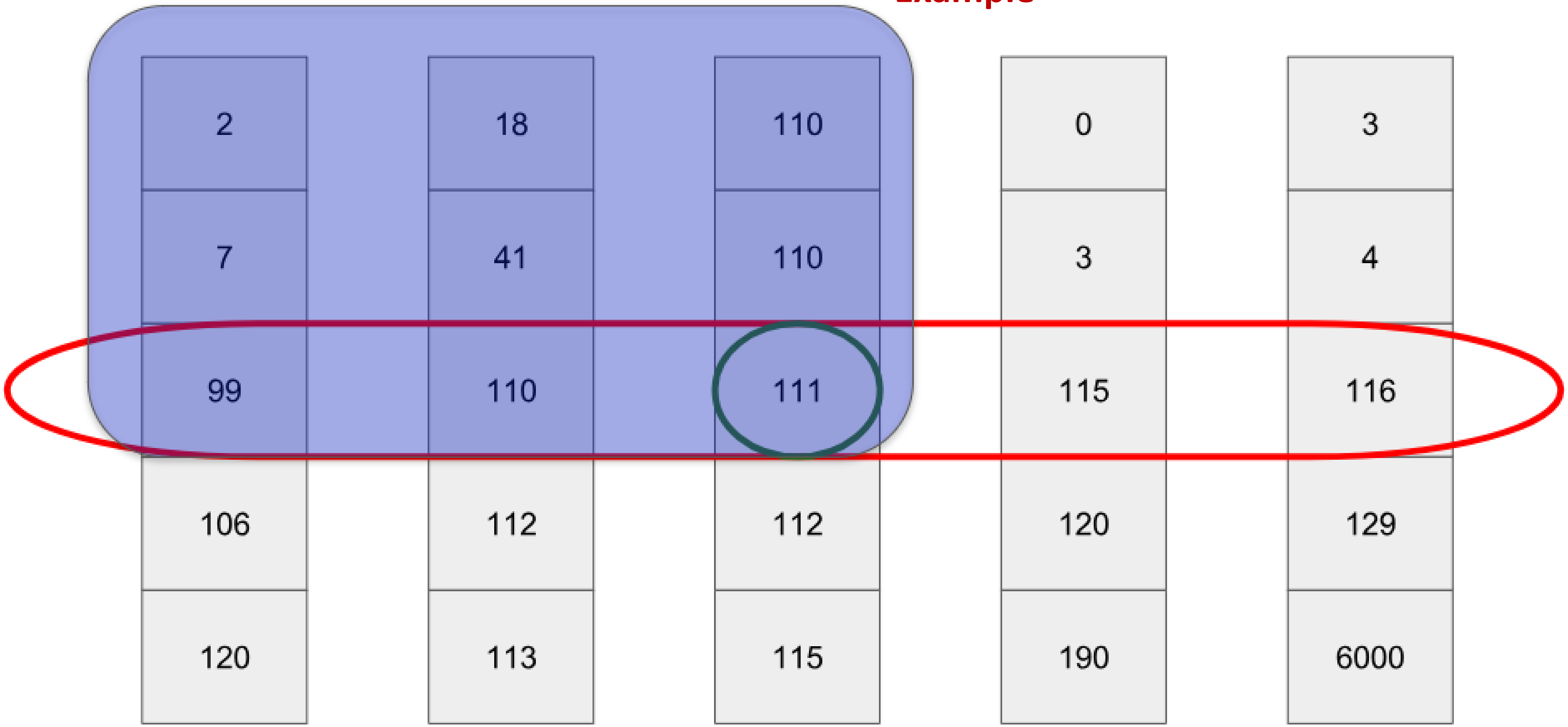




At least half the group medians are $\leq x$, which is at least $\lfloor \lfloor n/5 \rfloor /2 \rfloor = \lfloor n/10 \rfloor$ medians

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

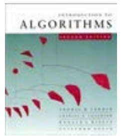
Example



There are 13 elements smaller than 111 and 11 elements larger than 111. **Rank of 111 is 14**.

<u>Median-of-medians:</u> guarantees that at least (1/2) * (3/5) * n elements can be discarded in the last phase of the median-finding algorithm. Thus, at most (7/10)*n elements survive in the last phase of the linear-time median finding algorithm.

Analysis: continued



Worst-case linear-time order statistics: Compute the i'th largest element of the array

Eg., the (n/2)'th element, the (log n)'th element, etc.

SELECT(i, n)

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 $n/5$ group medians to be the pivot.
3. Partition around the pivot x. Let k = rank(x).
4. **if** $i = k$ **then return** x

   elseif $i < k$

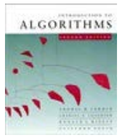
   then recursively SELECT the ith smallest element in the lower part

   else recursively SELECT the $(i–k)$th smallest element in the upper part

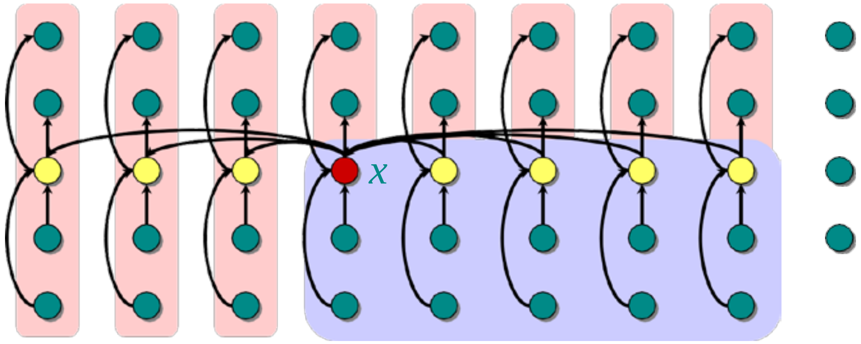
SELECT(t/2, t), where t=$\lfloor n/5 \rfloor$

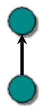
How many elements do we recurse on here?

Notation: Let T(n) be the worst-case running time for any ***i*** and inputs of size ***n***



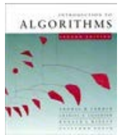
Analysis (Assume all elements are distinct.)





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

- Therefore, at least $3\lfloor n/10 \rfloor$ elements are $\leq x$.
- Similarly, at least $3\lfloor n/10 \rfloor$ elements are $\geq x$.
- We'll use a minor simplification



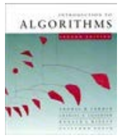
Minor simplification

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

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

- For $n \ge 50$, we have $3\lfloor n/10 \rfloor \ge n/4$.
- Therefore, for $n \ge 50$ the recursive call to SELECT in Step 4 is executed recursively on $\le n - n/4 = 3n/4$ 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 $T(n) = \Theta(1)$.

Analysis: continued



Developing the recurrence

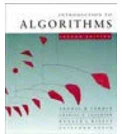
$T(n)$ SELECT(i, n) Recall: T(n) is independent of *i*

$\Theta(n)$ { 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 $k = \text{rank}(x)$ (that is, the position of x in the array after partitioning)

$T(3n/4)$ { **4. if** $i = k$ **then return** x
elseif $i < k$
then recursively SELECT the ith smallest element in the lower part
else recursively SELECT the $(i–k)$th smallest element in the upper part



Solving the recurrence

$$T(n) = T\left(\frac{n}{5}\right) + T\left(\frac{3n}{4}\right) + \Theta(n)$$

Induction:
$T(n) \le cn$

$$T(n) \le \frac{1}{5}cn + \frac{3}{4}cn + \Theta(n)$$
$$= \frac{19}{20}cn + \Theta(n)$$
$$= cn - (n/20 - \Theta(n))$$
$$\le cn,$$

if c is chosen large enough to handle both the $\Theta(n)$ and the base case.