

Order Statistics

- Maximum, Minimum $n-1$ comparisons

7	3	1	9	4	8	2	5	0	6
---	---	---	---	---	---	---	---	---	---

- MinMax
 - $2(n-1)$ comparisons
 - $3n/2$ comparisons
- Max and 2ndMax
 - $(n-1) + (n-2)$ comparisons
 - ???

k-Selection; Median

- Select the k -th smallest item in list
- Naïve Solution
 - Sort;
 - pick the k -th smallest item in sorted list.

$O(n \log n)$ time complexity
- Randomized solution: Average case $O(n)$
- Improved Solution: worst case $O(n)$

QuickSort

QUICKSORT(*array A, int p, int r*)

```
1  if ( $p < r$ )
2      then  $q \leftarrow \text{PARTITION}(A, p, r)$ 
3          QUICKSORT( $A, p, q - 1$ )
4          QUICKSORT( $A, q + 1, r$ )
```

To sort array call QUICKSORT($A, 1, \text{length}[A]$).

PARTITION(*array A, int p, int r*)

```
1   $x \leftarrow A[r]$  ▷ Choose pivot
2   $i \leftarrow p - 1$ 
3  for  $j \leftarrow p$  to  $r - 1$ 
4      do if ( $A[j] \leq x$ )
5          then  $i \leftarrow i + 1$ 
6              exchange  $A[i] \leftrightarrow A[j]$ 
7  exchange  $A[i + 1] \leftrightarrow A[r]$ 
8  return  $i + 1$ 
```

Page 146, CLRS

RandomizedPartition

- RandomizedPartition picks the pivot uniformly at random from among the elements in the list to be partitioned.

Homework

- **Statement of Collaboration**

- Take it **seriously**.
- **Reproduce** the statement faithfully and sign it by hand.
- For each problem, explain **separately** the sources and your collaborations with other people.
- Your homework **will not be graded** without the signed statement.

- **Extra Credit Problem**

- You can turn it in any time until second last class day (Dec 4th).
- You may retry a problem, but don't waste my time.
- You will not get partial credit on an extra credit problem.
- Put it on a separate sheet of paper and label it appropriately.

Announcements

- MidTerm Exam 1: October 16 in class
- MidTerm Exam 2: Last day of class
- Final: NO FINAL EXAM

QuickSelect: a variant of QuickSort

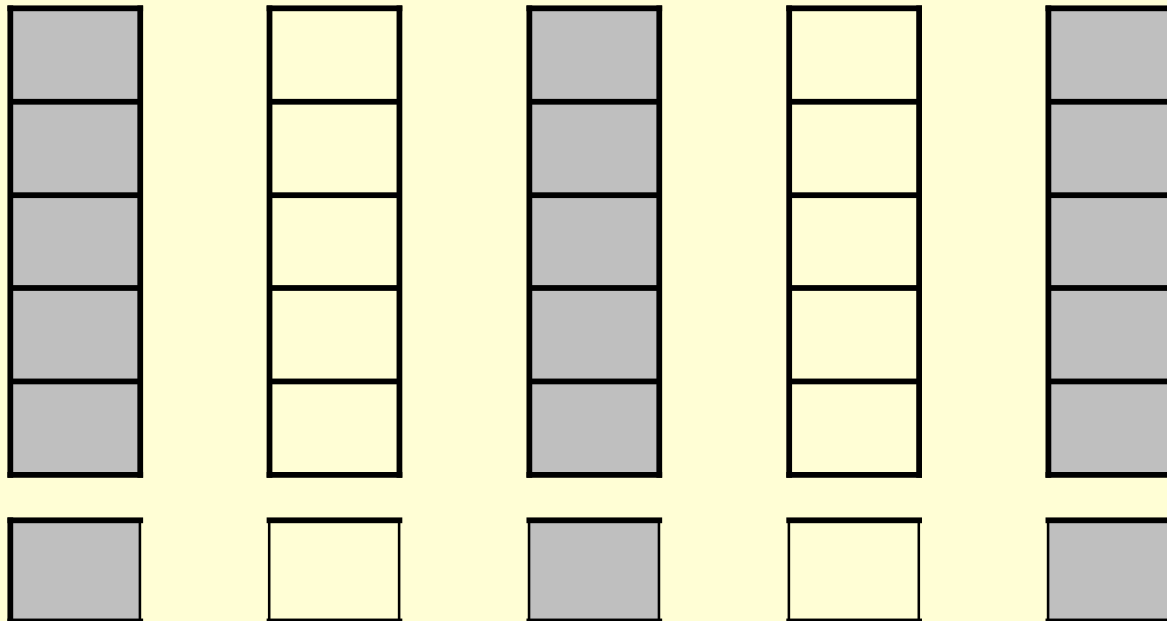
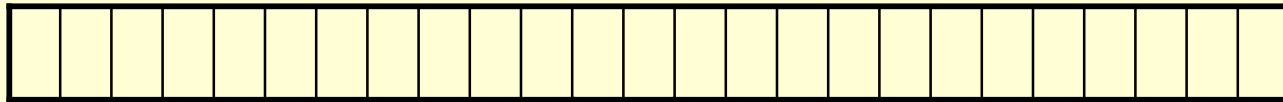
QUICKSELECT(*array A, int k, int p, int r*)

▷ Select k -th largest in subarray $A[p..r]$

```
1  if ( $p = r$ )
2      then return  $A[p]$ 
3   $q \leftarrow$  PARTITION( $A, p, r$ )
4   $i \leftarrow q - p + 1$     ▷ Compute rank of pivot
5  if ( $i = k$ )
6      then return  $A[q]$ 
7  if ( $i > k$ )
8      then return QUICKSELECT( $A, k, p, q$ )
9  else return QUICKSELECT( $A, k - i, q + 1, r$ )
```

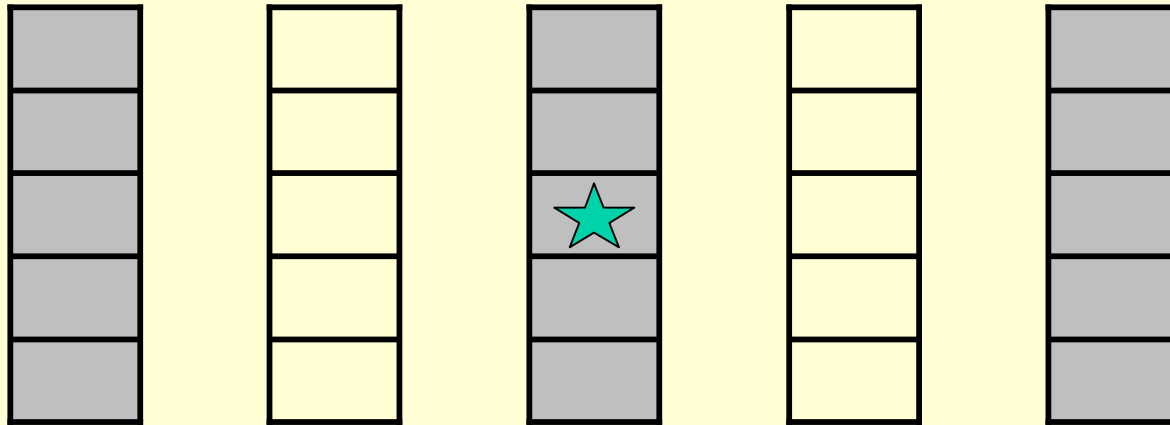
k-Selection & Median: Improved Algorithm

- Start with initial array



k-Selection & Median: Improved Algorithm(Cont'd)

- Use median of medians as pivot



- $T(n) < O(n) + T(n/5) + T(3n/4)$

ImprovedSelect

IMPROVEDSELECT(*array A, int k, int p, int r*)

▷ Select k -th largest in subarray $A[p..r]$

```
1  if ( $p = r$ )
2    then return  $A[p]$ 
3    else  $N \leftarrow r - p + 1$ 
4    Partition  $A[p..r]$  into subsets of 5 elements and
   collect all medians of subsets in  $B[1..\lceil N/5 \rceil]$ .
5     $Pivot \leftarrow$  IMPROVEDSELECT( $B, 1, \lceil N/5 \rceil, \lceil N/10 \rceil$ )
6     $q \leftarrow$  PIVOTPARTITION( $A, p, r, Pivot$ )
7     $i \leftarrow q - p + 1$     ▷ Compute rank of pivot
8    if ( $i = k$ )
9      then return  $A[q]$ 
10   if ( $i > k$ )
11     then return IMPROVEDSELECT( $A, k, p, q - 1$ )
12     else return IMPROVEDSELECT( $A, k - i, q + 1, r$ )
```

PivotPartition

PIVOTPARTITION(*array A, int p, int r, item Pivot*)

▷ Partition using provided *Pivot*

```
1  $i \leftarrow p - 1$ 
2 for  $j \leftarrow p$  to  $r$ 
3     do if ( $A[j] \leq Pivot$ )
4         then  $i \leftarrow i + 1$ 
5             exchange  $A[i] \leftrightarrow A[j]$ 
6 return  $i + 1$ 
```