

Lecture slides by Kevin Wayne
Copyright © 2005 Pearson-Addison Wesley
Copyright © 2013 Kevin Wayne

http://www.cs.princeton.edu/~wayne/kleinberg-tardos

# 5. DIVIDE AND CONQUER I

quickselect demo

# 3-way partition array so that:

- Pivot element *p* is in place.
- Smaller elements in left subarray *L*.
- Equal elements in middle subarray M.
- Larger elements in right subarray *R*.

Recur in one subarray—the one containing the  $k^{th}$  smallest element.

select the  $k = 8^{th}$  smallest

# 3-way partition array so that:

- Pivot element *p* is in place.
- Smaller elements in left subarray *L*.
- Equal elements in middle subarray M.
- Larger elements in right subarray *R*.

Recur in one subarray—the one containing the  $k^{th}$  smallest element.

#### choose a pivot element at random and partition

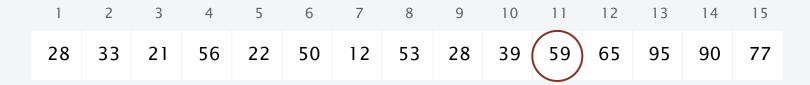


# 3-way partition array so that:

- Pivot element *p* is in place.
- Smaller elements in left subarray *L*.
- Equal elements in middle subarray *M*.
- Larger elements in right subarray *R*.

Recur in one subarray—the one containing the  $k^{th}$  smallest element.

#### partitioned array

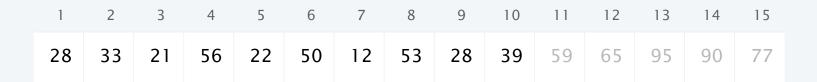


# 3-way partition array so that:

- Pivot element *p* is in place.
- Smaller elements in left subarray *L*.
- Equal elements in middle subarray M.
- Larger elements in right subarray *R*.

Recur in one subarray—the one containing the  $k^{th}$  smallest element.

#### recursively select 8th smallest element in left subarray



# 3-way partition array so that:

- Pivot element *p* is in place.
- Smaller elements in left subarray *L*.
- Equal elements in middle subarray M.
- Larger elements in right subarray *R*.

Recur in one subarray—the one containing the  $k^{th}$  smallest element.

#### choose a pivot element at random and partition

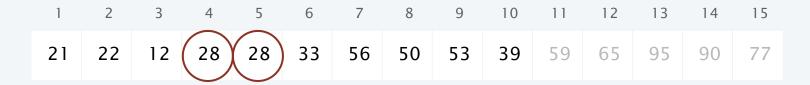


# 3-way partition array so that:

- Pivot element *p* is in place.
- Smaller elements in left subarray *L*.
- Equal elements in middle subarray *M*.
- Larger elements in right subarray *R*.

Recur in one subarray—the one containing the  $k^{th}$  smallest element.

#### partitioned array

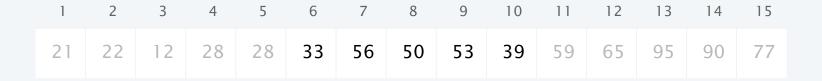


# 3-way partition array so that:

- Pivot element *p* is in place.
- Smaller elements in left subarray *L*.
- Equal elements in middle subarray M.
- Larger elements in right subarray *R*.

Recur in one subarray—the one containing the  $k^{th}$  smallest element.

#### recursively select the 3<sup>rd</sup> smallest element in right subarray



 $k = 3^{rd}$  smallest

# 3-way partition array so that:

- Pivot element *p* is in place.
- Smaller elements in left subarray *L*.
- Equal elements in middle subarray *M*.
- Larger elements in right subarray *R*.

Recur in one subarray—the one containing the  $k^{th}$  smallest element.

#### choose a pivot element at random and partition



 $k = 3^{rd}$  smallest

# 3-way partition array so that:

- Pivot element *p* is in place.
- Smaller elements in left subarray *L*.
- Equal elements in middle subarray *M*.
- Larger elements in right subarray *R*.

Recur in one subarray—the one containing the  $k^{th}$  smallest element.

#### partitioned array



 $k = 3^{rd}$  smallest

# 3-way partition array so that:

- Pivot element *p* is in place.
- Smaller elements in left subarray *L*.
- Equal elements in middle subarray *M*.
- Larger elements in right subarray *R*.

Recur in one subarray—the one containing the  $k^{th}$  smallest element.

stop: desired element is in middle subarray

