# Decrease & Conquer

# Variable Size Decrease

- The size-reduction pattern varies from one iteration of algorithm to another.

**Example**: Euclid's algorithm for computing the $\gcd(m, n)$

$$\gcd(m, n) = \gcd(n, m \bmod n)$$

# Selection Problem

**The selection problem:** Find the $k$th smallest element in the list of $n$ numbers. This number is called $k$th order statistic.

- $k = 1$: find the **smallest element**
- $k = n$: find the **largest element**
- $k = \lfloor n/2 \rfloor$: find the **median**

**Example:** Given $4, 1, 10, 9, 7, 12, 8, 2, 15$. **Median?**

# Selection Problem

- We can find the $k$th smallest element in a list by sorting the list first, and then selecting the $k$th element.
- The time of such an algorithm is determined by the efficiency of the sorting algorithm used.
- For instance, mergesort: $O(n \log n)$
- We can take advantage of the idea of partitioning a given list **around some value** $p$.

# Partitioning

**Partitioning** a list around some **value** $p$ (pivot) is a rearrangement of the list's elements so that

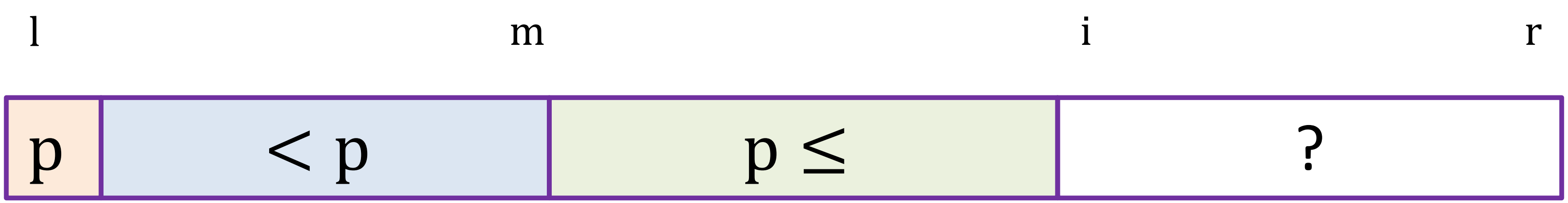
- the left part contains all the elements $\leq p$
- followed by the pivot $p$ itself
- the right part contains all elements $\geq p$.

| | | |
|---|---|---|
| $\leq p$ | $p$ | $p \leq$ |

# Lomuto Partitioning

A subarray $A[l..r]$ $(0 \leq l < r \leq n)$ is composed of three contiguous segments followed pivot $p$:

- a segment with elements **smaller** than $p$
- a segment with elements **greater** than or equal to $p$
- a segment with elements yet to be **compared** to $p$



# Lomuto Partitioning

1. Start with $i = l + 1$, scan the subarray $A[l..r]$ left to right.

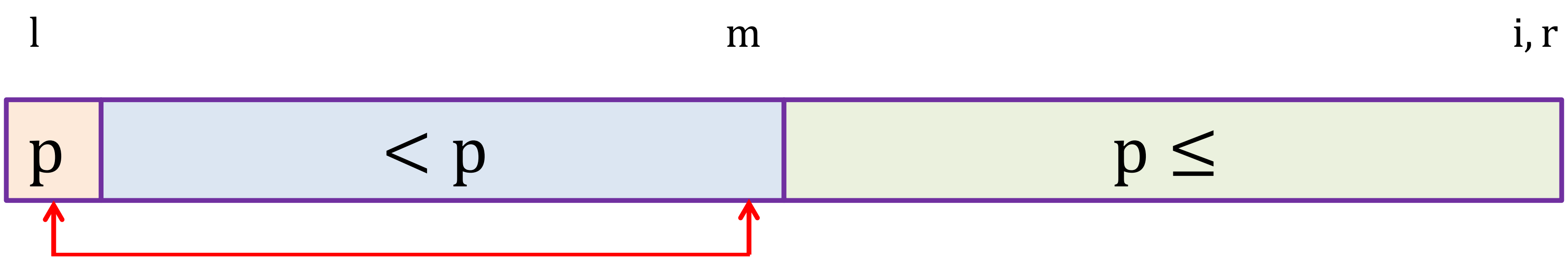
2. On each iteration, compare first element $A[i]$ in the third segment with the pivot $p$.

3. If $A[i] \geq p$, **increment** $i$ to expand the second segment, shrinking the third segment.

4. If $A[i] < p$, the first segment needs to be expanded: increment $m$, the index of the last element of the first segment, and **swap** $A[i]$ and $A[m]$, then increment $i$ to point the first element in the third segment.

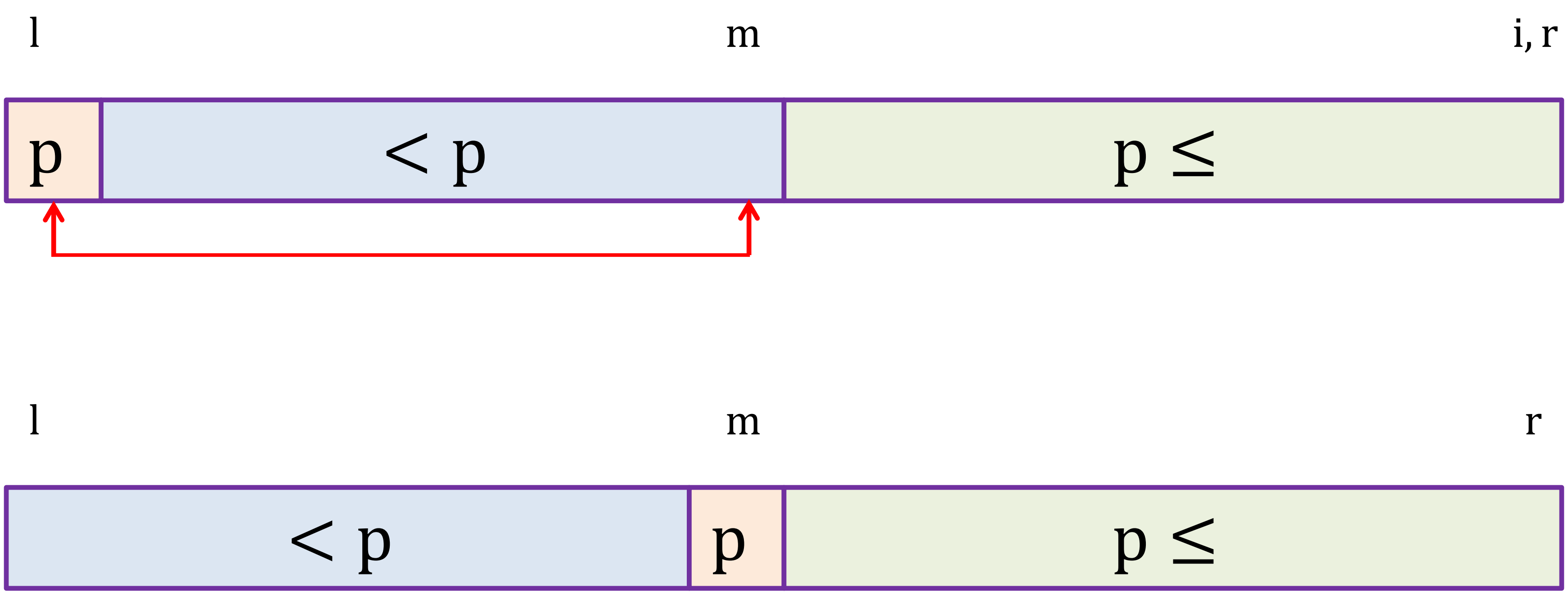
# Lomuto Partitioning

5. After no unprocessed element remain, swap $A[m]$ with the pivot $p$ to finish the partitioning.



# Lomuto Partitioning

5. After no unprocessed element remain, **swap** $A[m]$ with the pivot $p$ to finish the partitioning.



# Lomuto Partitioning

**Algorithm** LomutoPartition($A[l..r]$)

// Input: A subarray $A[l..r]$ of $A[0..n]$

//Output: Partition of $A[l..r]$ and new position of $p$

$p \leftarrow A[l]; \quad m \leftarrow l$

**for** $i \leftarrow l + 1$ **to** $r$ **do**

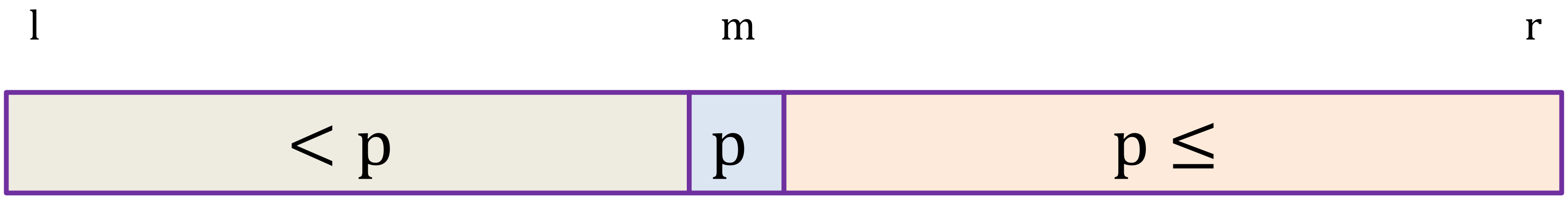
  **if** $A[i] < p$

    $m \leftarrow m + 1$; swap($A[m], A[i]$)

swap($A[l], A[m]$)

**return** $m$

# Quickselect




1. If $m = k - 1$, the pivot $p$ is the $k$th smallest element
2. If $m > k - 1$, $k$th smallest element can be found in the left part
3. If $m < k - 1$, $k$th smallest element can be found as the $(k - m)$th smallest element in the right part

# Quickselect

**Algorithm** Quickselect($A[l..r], k$)

// Input: A subarray $A[l..r]$ of $A[0..n]$

//Output: The value of kth smallest element in $A[l..r]$

$m \leftarrow$ LomutoPartition($A[l..r]$)

**if** $m = k - 1$ **return** $A[m]$

**else if** $m > k - 1$

Quickselect($A[l..m-1], k$)

**else** Quickselect($A[m+1..r], k - 1 - m$)

# Quickselect

**Example:** 4, 1, 10, 8, 7, 12, 9, 2, 15, $k = 5$

| l,m | i | | | | | | | r |
|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| 4 | 1 | 10 | 8 | 7 | 12 | 9 | 2 | 15 |

# Quickselect

**Example:** 4, 1, 10, 8, 7, 12, 9, 2, 15, $k = 5$

| l | m,i | | | | | | | r |
|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| 4 | 1 | 10 | 8 | 7 | 12 | 9 | 2 | 15 |

# Quickselect

**Example:** 4, 1, 10, 8, 7, 12, 9, 2, 15, $k = 5$

| l | m | i | | | | | | r |
|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| 4 | 1 | 10 | 8 | 7 | 12 | 9 | 2 | 15 |

# Quickselect

**Example:** 4, 1, 10, 8, 7, 12, 9, 2, 15, $k = 5$

| l | m | i | | | | | | r |
|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| 4 | 1 | 10 | 8 | 7 | 12 | 9 | 2 | 15 |

# Quickselect

**Example:** 4, 1, 10, 8, 7, 12, 9, 2, 15, $k = 5$

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| l | m | | i | | | | | r |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| 4 | 1 | 10 | 8 | 7 | 12 | 9 | 2 | 15 |

# Quickselect

**Example:** 4, 1, 10, 8, 7, 12, 9, 2, 15, $k = 5$

| l | m | | i | | | | | r |
|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| 4 | 1 | 10 | 8 | 7 | 12 | 9 | 2 | 15 |

# Quickselect

**Example:** 4, 1, 10, 8, 7, 12, 9, 2, 15, $k = 5$

| l | m | | | i | | | | r |
|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| 4 | 1 | 10 | 8 | 7 | 12 | 9 | 2 | 15 |

# Quickselect

**Example:** 4, 1, 10, 8, 7, 12, 9, 2, 15, $k = 5$

| l | m | | | i | | | | r |
|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| 4 | 1 | 10 | 8 | 7 | 12 | 9 | 2 | 15 |

# Quickselect

**Example:** $4,\ 1,\ 10,\ 8,\ 7,\ 12,\ 9,\ 2,\ 15,\quad k = 5$

| l | m | | | | i | | | r |
|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| 4 | 1 | 10 | 8 | 7 | 12 | 9 | 2 | 15 |

# Quickselect

**Example:** $4,\ 1,\ 10,\ 8,\ 7,\ 12,\ 9,\ 2,\ 15,\qquad k = 5$

| l | m | | | | i | | | r |
|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| 4 | 1 | 10 | 8 | 7 | 12 | 9 | 2 | 15 |

# Quickselect

**Example:** 4, 1, 10, 8, 7, 12, 9, 2, 15, $k = 5$

| l | m | | | | | i | | r |
|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| 4 | 1 | 10 | 8 | 7 | 12 | 9 | 2 | 15 |

# Quickselect

**Example:** 4, 1, 10, 8, 7, 12, 9, 2, 15, $k = 5$

| l | m | | | | | i | | r |
|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| 4 | 1 | 10 | 8 | 7 | 12 | 9 | 2 | 15 |

# Quickselect

**Example:** 4, 1, 10, 8, 7, 12, 9, 2, 15, $k = 5$

| l | m | | | | | | i | r |
|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| 4 | 1 | 10 | 8 | 7 | 12 | 9 | 2 | 15 |

# Quickselect

**Example:** 4, 1, 10, 8, 7, 12, 9, 2, 15, $k = 5$

| l | | m | | | | | i | r |
|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| 4 | 1 | 10 | 8 | 7 | 12 | 9 | 2 | 15 |

# Quickselect

**Example:** 4, 1, 10, 8, 7, 12, 9, 2, 15, $k = 5$

| l | | m | | | | | i | r |
|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| 4 | 1 | 2 | 8 | 7 | 12 | 9 | 10 | 15 |

# Quickselect

**Example:** 4, 1, 10, 8, 7, 12, 9, 2, 15, $k = 5$

| l | | m | | | | | | i, r |
|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| 4 | 1 | 2 | 8 | 7 | 12 | 9 | 10 | 15 |

# Quickselect

**Example:** 4, 1, 10, 8, 7, 12, 9, 2, 15, $k = 5$

| l | | m | | | | | | i, r |
|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| 4 | 1 | 2 | 8 | 7 | 12 | 9 | 10 | 15 |

# Quickselect

**Example:** $4,\ 1,\ 10,\ 8,\ 7,\ 12,\ 9,\ 2,\ 15,\qquad k = 5$

| l | | m | | | | | | i, r |
|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| 2 | 1 | 4 | 8 | 7 | 12 | 9 | 10 | 15 |

# Quickselect

**Example:** 4, 1, 10, 8, 7, 12, 9, 2, 15, $k = 5$

| l | | m | | | | | | r |
|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| 2 | 1 | 4 | 8 | 7 | 12 | 9 | 10 | 15 |

$$2 < 5 - 1 = 4$$

# Quickselect

**Example:** 4, 1, 10, 8, 7, 12, 9, 2, 15, $k = 5$

| | | | $l, m$ | $i$ | | | | $r$ |
|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| 2 | 1 | 4 | 8 | 7 | 12 | 9 | 10 | 15 |

# Quickselect

**Example:** 4, 1, 10, 8, 7, 12, 9, 2, 15, $k = 5$

| | | | l | m, i | | | | r |
|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| 2 | 1 | 4 | 8 | 7 | 12 | 9 | 10 | 15 |

# Quickselect

**Example:** 4, 1, 10, 8, 7, 12, 9, 2, 15, $k = 5$

| | | | l | m | i | | | r |
|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| 2 | 1 | 4 | 8 | 7 | 12 | 9 | 10 | 15 |

# Quickselect

**Example:** 4, 1, 10, 8, 7, 12, 9, 2, 15, $k = 5$

| | | | l | m | i | | | r |
|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| 2 | 1 | 4 | 8 | 7 | 12 | 9 | 10 | 15 |

# Quickselect

**Example:** 4, 1, 10, 8, 7, 12, 9, 2, 15, $k = 5$

| | | | l | m | | i | | r |
|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| 2 | 1 | 4 | 8 | 7 | 12 | 9 | 10 | 15 |

# Quickselect

**Example:** 4, 1, 10, 8, 7, 12, 9, 2, 15, $k = 5$

| | | | l | m | | i | | r |
|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| 2 | 1 | 4 | 8 | 7 | 12 | 9 | 10 | 15 |

# Quickselect

**Example:** 4, 1, 10, 8, 7, 12, 9, 2, 15, $k = 5$

| | | | l | m | | | i | r |
|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| 2 | 1 | 4 | 8 | 7 | 12 | 9 | 10 | 15 |

# Quickselect

**Example:** 4, 1, 10, 8, 7, 12, 9, 2, 15, $k = 5$

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | | l | m | | | i | r |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| 2 | 1 | 4 | 8 | 7 | 12 | 9 | 10 | 15 |

# Quickselect

**Example:** 4, 1, 10, 8, 7, 12, 9, 2, 15, $k = 5$

| | | | l | m | | | | i, r |
|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| 2 | 1 | 4 | 8 | 7 | 12 | 9 | 10 | 15 |

# Quickselect

**Example:** 4, 1, 10, 8, 7, 12, 9, 2, 15, $k = 5$

| | | | l | m | | | | i, r |
|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| 2 | 1 | 4 | 8 | 7 | 12 | 9 | 10 | 15 |

# Quickselect

**Example:** $4,\ 1,\ 10,\ 8,\ 7,\ 12,\ 9,\ 2,\ 15,$ $k = 5$

| | | | l | m | | | | i, r |
|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| 2 | 1 | 4 | 7 | 8 | 12 | 9 | 10 | 15 |

$$4 = 5 - 1$$

# Quickselect

**Example:** 4, 1, 10, 8, 7, 12, 9, 2, 15, $k = 5$

m

| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|---|
| 2 | 1 | 4 | 7 | 8 | 12 | 9 | 10 | 15 |

# Quickselect

**The efficiency of Quickselect:**

- **best case:** partitioning an $n$-element array always requires $n-1$ comparison

$$C_b(n) = n - 1 \in \Theta(n)$$

- **worst case:** one part always empty and the other part contains $n-1$ elements, which can happen on each of the $n-1$ iterations; if $k = n$:

$$C_w(n) = (n-1) + (n-2) + \cdots + 1 = \frac{n(n-1)}{2} \in \Theta(n^2)$$

# Exercise

- Apply quickselect to find the median of the list of numbers:

$$9, 12, 5, 17, 20, 30, 8$$