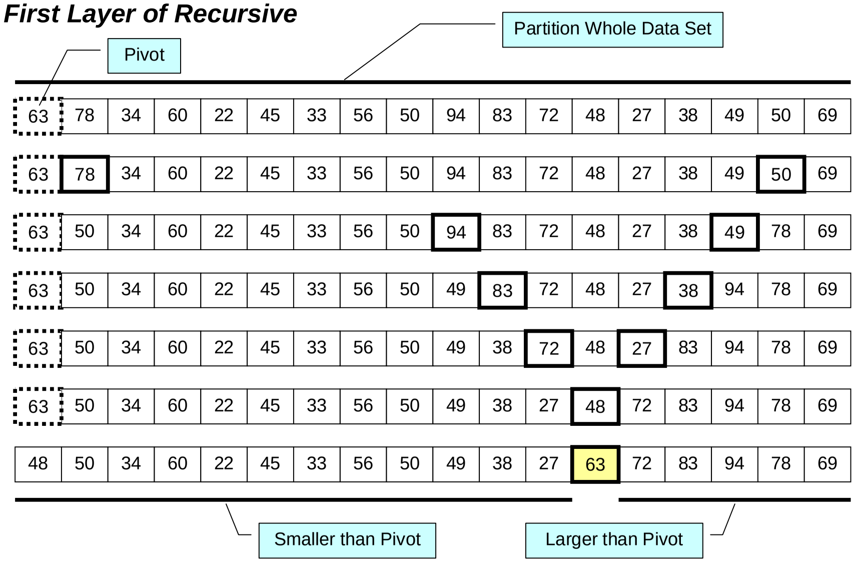
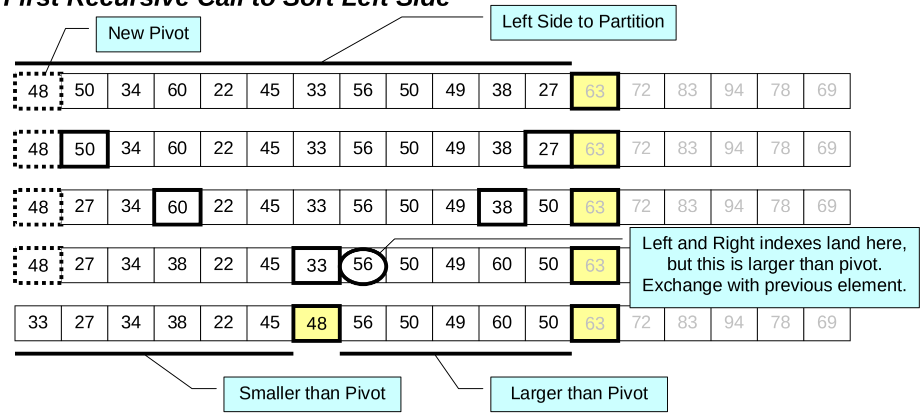
CST8130: Data Structures
Quicksort: A Recursive Sort Algorithm

First Layer of Recursive



63	78	34	60	22	45	33	56	50	94	83	72	48	27	38	49	50	69
63	78	34	60	22	45	33	56	50	94	83	72	48	27	38	49	50	69
63	50	34	60	22	45	33	56	50	94	83	72	48	27	38	49	78	69
63	50	34	60	22	45	33	56	50	49	83	72	48	27	38	94	78	69
63	50	34	60	22	45	33	56	50	49	38	72	48	27	83	94	78	69
63	50	34	60	22	45	33	56	50	49	38	27	48	72	83	94	78	69
48	50	34	60	22	45	33	56	50	49	38	27	63	72	83	94	78	69

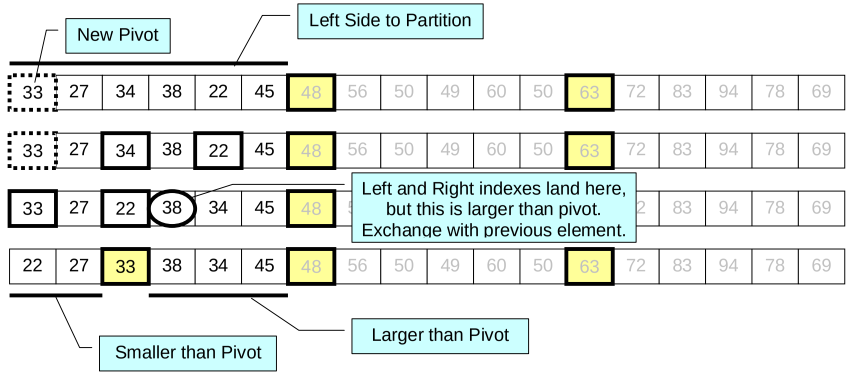
First Recursive Call to Sort Left Side



Left and Right indexes land here, but this is larger than pivot. Exchange with previous element.

48	50	34	60	22	45	33	56	50	49	38	27	63	72	83	94	78	69
48	50	34	60	22	45	33	56	50	49	38	27	63	72	83	94	78	69
48	27	34	60	22	45	33	56	50	49	38	50	63	72	83	94	78	69
48	27	34	38	22	45	33	56	50	49	60	50	63					
33	27	34	38	22	45	48	56	50	49	60	50	63	72	83	94	78	69

Second Recursive Call to Sort Left Side



New Pivot

Left Side to Partition

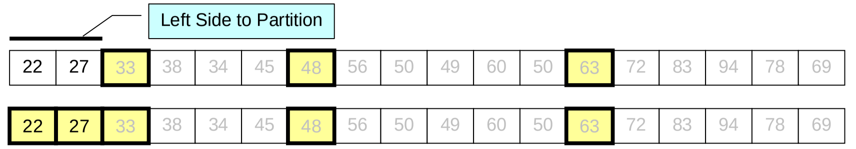
33	27	34	38	22	45	48	56	50	49	60	50	63	72	83	94	78	69
33	27	34	38	22	45	48	56	50	49	60	50	63	72	83	94	78	69
33	27	22	38	34	45	48	5						2	83	94	78	69
22	27	33	38	34	45	48	56	50	49	60	50	63	72	83	94	78	69

Left and Right indexes land here, but this is larger than pivot. Exchange with previous element.

Smaller than Pivot

Larger than Pivot

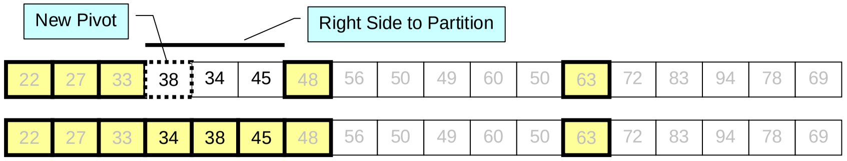
Third Recursive Call to Sort Left Side



Left Side to Partition

22	27	33	38	34	45	48	56	50	49	60	50	63	72	83	94	78	69
22	27	33	38	34	45	48	56	50	49	60	50	63	72	83	94	78	69

First Recursive Call to Sort Right Side

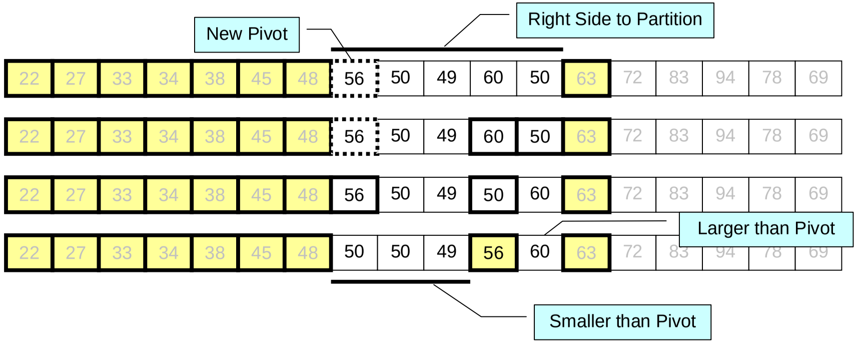


New Pivot

Right Side to Partition

22	27	33	38	34	45	48	56	50	49	60	50	63	72	83	94	78	69
22	27	33	34	38	45	48	56	50	49	60	50	63	72	83	94	78	69

Second Recursive Call to Sort Right Side



New Pivot

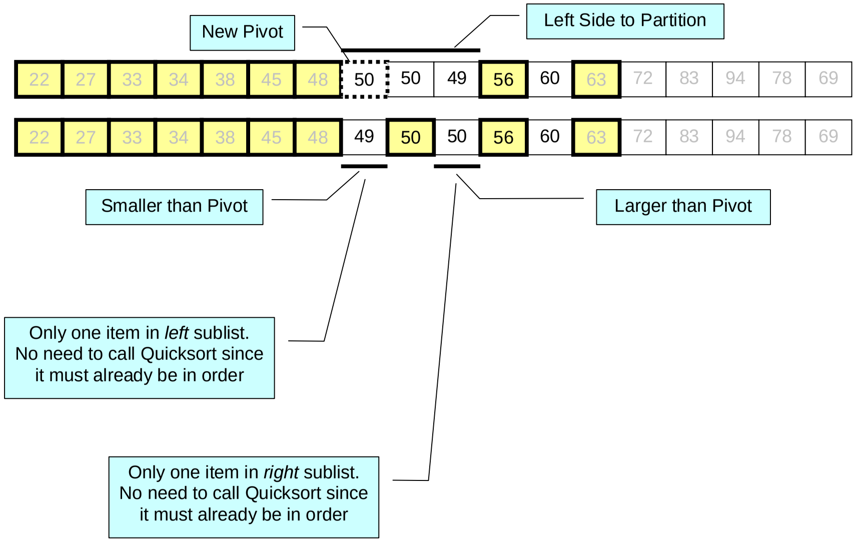
Right Side to Partition

22	27	33	34	38	45	48	56	50	49	60	50	63	72	83	94	78	69
22	27	33	34	38	45	48	56	50	49	60	50	63	72	83	94	78	69
22	27	33	34	38	45	48	56	50	49	50	60	63	72	83	94	78	69
22	27	33	34	38	45	48	50	50	49	56	60	63	72	83	94	78	69

Larger than Pivot

Smaller than Pivot

Forth Recursive Call to Sort Left Side



Only one item in *left* sublist.
No need to call Quicksort since
it must already be in order

Only one item in *right* sublist.
No need to call Quicksort since
it must already be in order

Algorithm Overview

- **Find the pivot, and partition the data set.**
- ***Recursively call quicksort* to sort left side of pivot, then right side of pivot.**

Efficiency: $O(n \log_2 n)$

Find the Pivot and Partition the Data Set

- **Start with given data set having indexes ranging from *start* to *end***
- **Select index of first element as *pivot***
- **Create two index counters, *left* (at *start* + 1) and *right* (at *end*)**
- **Scan and exchange (detailed in box) while *left* index is less than *right* index**
  - **Find *left* item *larger* than pivot (while *left* index is less than *right* index)**
  - **Find *right* item *smaller* than pivot (while *left* index is less than *right* index)**
  - **if left and right indexes have not met (that is, we’re still partitioning)**

    Swap *left* and *right* items

    increment *left* and decrement *right*
- **Move the pivot to its correct final position**
  - **if value at the *left* index is larger than *pivot***

    exchange value at *pivot* location with value at location at *left* - 1
  - **else**

    exchange value at *pivot* location with value at location at *left*

Recursively sort left subset of data and right subset of data