

Tinker Academy

Programming Using Java
(Analysis Bubble Sort)

Bubble Sort

Simple Sorting Algorithm

5 1 2 3 4

Unsorted Array

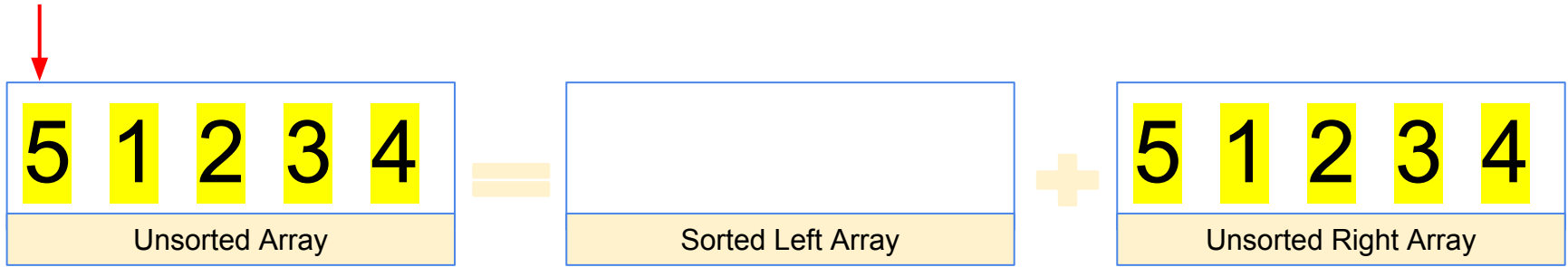
1 2 3 4 5

Unsorted Array

Sorting is "slow", which means it takes more cpu time to complete sort

Bubble Sort

Sorts array progressively from left to right using Pivot

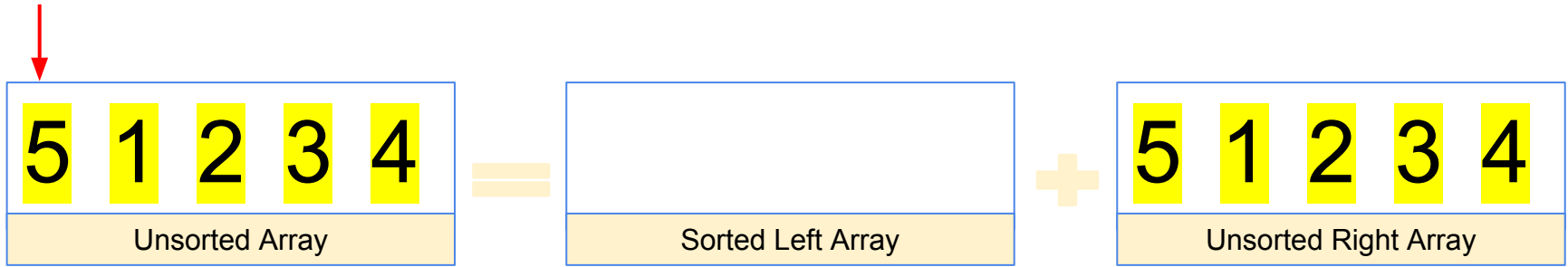


Pivot divides the array into 2, a sorted left subarray, unsorted right array

Pivot at index 0 divides array into empty left subarray, full right subarray

Bubble Sort

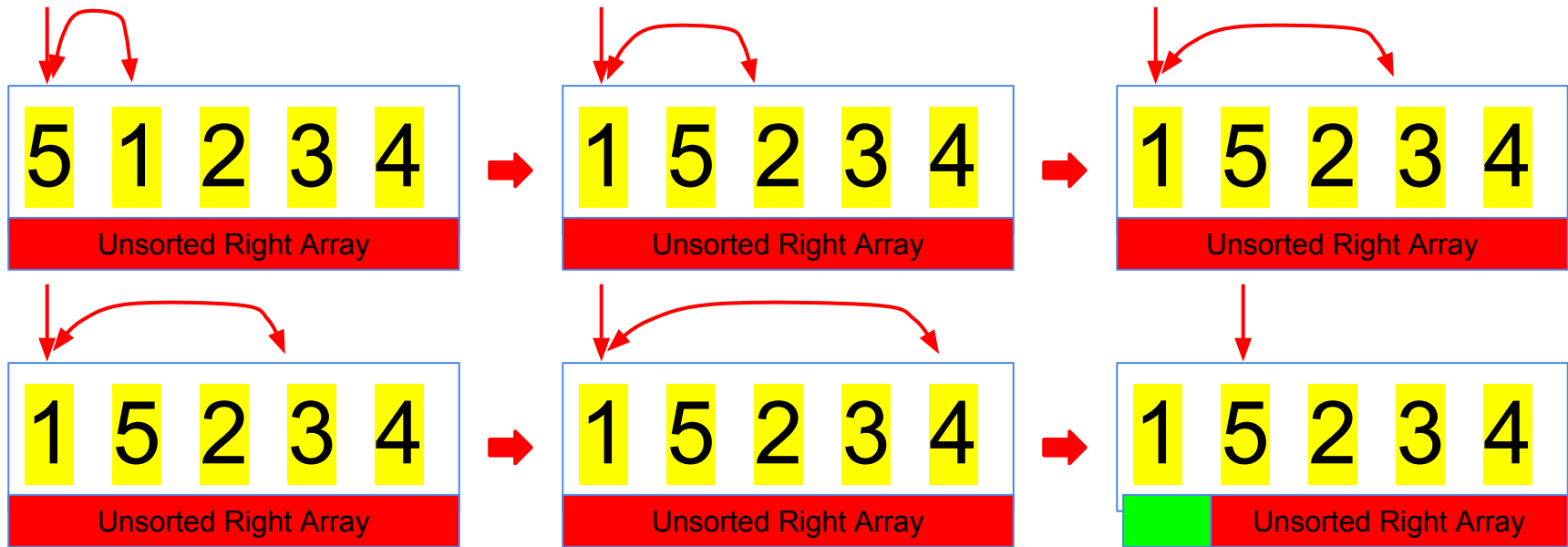
Bubble Sort first ensures left subarray always sorted



When **Pivot** at index 0, left subarray is empty and is sorted by definition

Bubble Sort

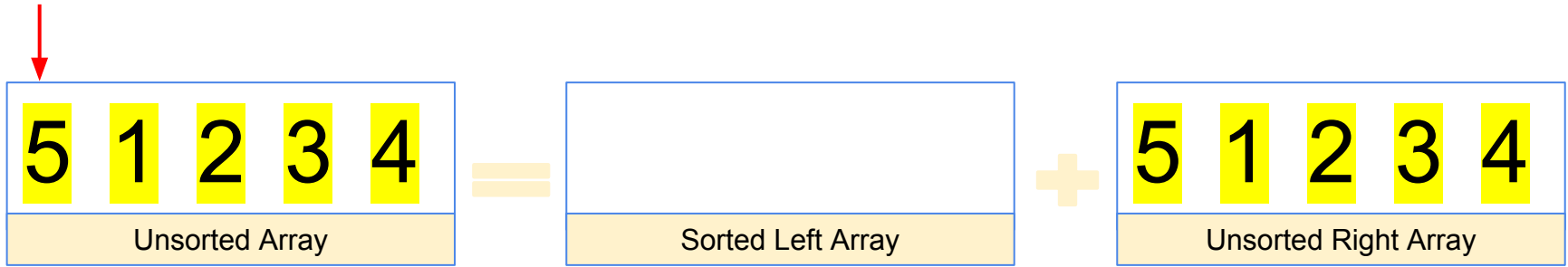
Bubble Sort next ensures item at pivot index is lowest in right subarray



Swaps item if item to right of Pivot < Pivot Item

Bubble Sort

Bubble Sort Requires Comparisons

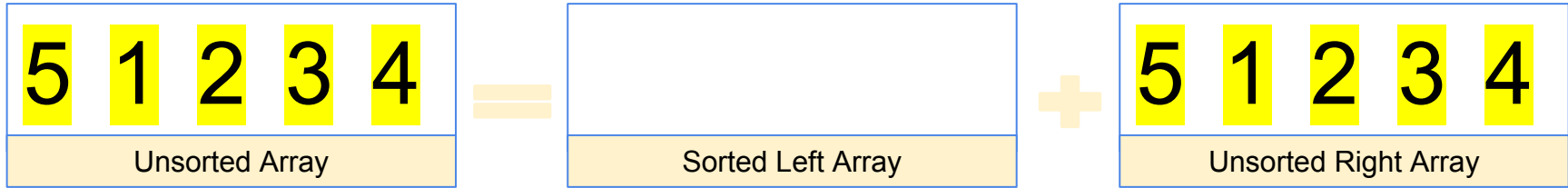


For each **Pivot**, each item to right compared with pivot item

For 5 item array, when **Pivot** at index 0, 4 comparisons

Bubble Sort

For 5 item array, when **Pivot** at index 0, 4 comparisons



For 5 item array, when **Pivot** at index 1, 3 comparisons

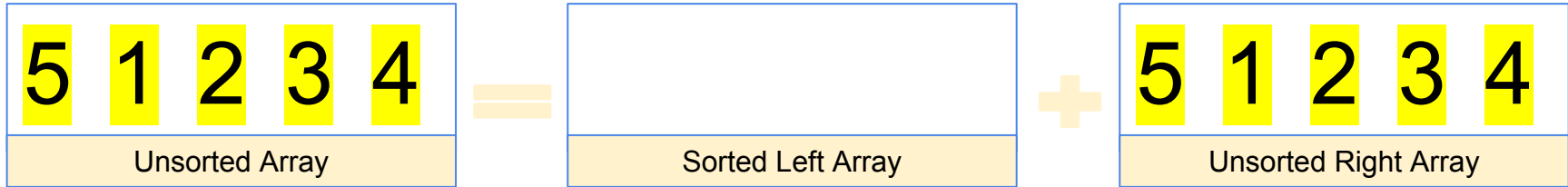
For 5 item array, when **Pivot** at index 2, 2 comparisons

For 5 item array, when **Pivot** at index 3, 1 comparisons

For 5 item array, when **Pivot** at index 4, 0 comparisons

Bubble Sort

Total Comparisons = $(4) + (3) + (2) + (1) + (0)$



In General, for array of size n

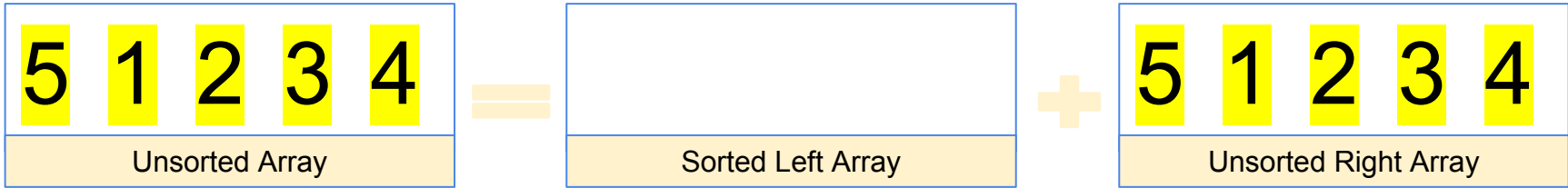
Total Comparisons = $(n-1) + (n-2) + (n-3) + \dots + (0) = n(n-1)/2$

For array of size n , # of calculations almost n^2

As n is large, for example 1 Million, # Comparisons \approx 1 Trillion!

Bubble Sort

Array of size 1 Million, # Comparisons \approx 1 Trillion!



Thats too many comparisons!

Better sorting algorithms will reduce the number to \approx 20 million

20 million \ll 1 Trillion!