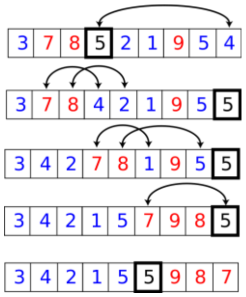
By: Alexis Tao, Ethan Tam, Marcus Ng, William Chen

Quick Sort Information Handout

Visual Exemplar



3 7 8 5 2 1 9 5 4

3 7 8 4 2 1 9 5 5

3 4 2 7 8 1 9 5 5

3 4 2 1 5 7 9 8 5

3 4 2 1 5 5 9 8 7

Method

- Select ____________
- Compare all elements of the array until all elements greater(__________) or smaller(___________) than the pivot are on the left and all elements smaller(_________) or greater(________) than the pivot are on the right
- Place the pivot on the wall, partitioning the array into two subarrays
- Apply quick sort ____________ to the sub arrays until all elements are sorted

NEW TERMINOLOGY	**Pivot:** An element of the array that is compared with all the other elements **Wall:** A divider used to partition an array or subarray in the quick sorting method **Current Element:** The element that is being compared to the pivot **Partition:** Dividing an array into different parts **Subarray:** A subsection of an array
EFFICIENCY	Picking the Pivot: • Last/First Element (__________ chance of worst case) • Randomly (__________ chance of worst case) • Median of the first, last, & middle elements (__________ worst case) Big O • Worst Case:____________ • Best Case: ____________

WHY	WHEN	WHERE
• Cache efficient & usually fast • High chance to pick a number that can ______________ the array into 2 parts	• No need for stable sort • Average __________is more important	• Used to sort arrays • Used for __________ arrays

PROS	CONS
• _________ method is easy to implement • On paper, more efficient than all other sorting methods • More memory efficient than merge sort • Best case sorting is O(nlogn) • Extremely efficient if right_________ is chosen • Able to deal with __________array sizes	• In the worst case, can be as slow as bubble sort (n^2) • Iterative implementation is difficult to implement (faster than recursive) • Is an ___________ sorting method (Does not keep things in relative order)