Lere's another idea for a sorting algorithm: Maintain a sorted subarray While(subarray is not full array) Find the smallest element remaining in the unsorted part. Insert it at the end of the sorted part.

7

Merge Sort Pseudocode MergeSort(input) { if (input.length == 1) return else smallerHalf = mergeSort(new [0, ..., mid]) largerHalf = mergeSort(new [mid + 1, ...]) return merge(smallerHalf, largerHalf) }

Swapping

How do we divide the array into "bigger than the pivot" and "less than the pivot?"

Swap the pivot to the far left.
 Make a pointer *i* on the left, and *j* on the right
 Until *i*, *j* meet

 While *A*[*i*] < pivot move *i* left
 While *A*[*j*] > pivot move *j* right

-Swap A[i], A[j]

4. Swap A[i] or A[i-1] with pivot.

22

Quick Sort Analysis (Take 1)
What is the best case and worst case for a pivot? -Best case: -Worst case:
Recurrences:
Best:
Worst:
Running times: -Best: -Worst: