

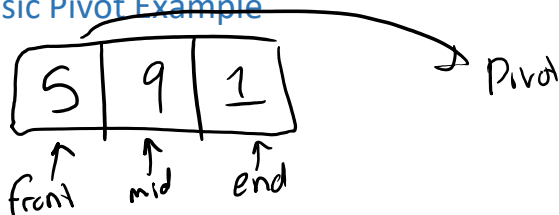
2018-11-15 QuickSort

Thursday, November 15, 2018 2:59 PM

Quicksort

- General idea: find a "pivot." Put items smaller than the pivot to the "left" of the pivot and larger items to the "right" of the pivot.
- Recursively do this on the subarray bound by the pivot
- There are several ways that we might select a pivot.
- Richard Weiss suggests "best of 3" approach:
 - Pick middle-most value between front element, middle element, and end element
 - We do this in order to avoid QuickSort's worst-case N^2 runtime

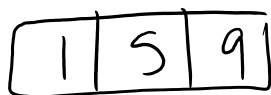
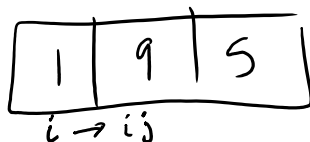
Basic Pivot Example



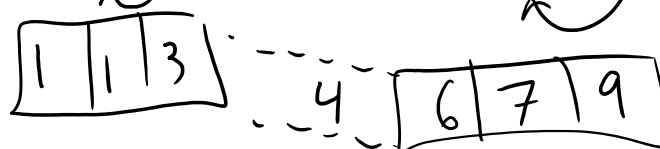
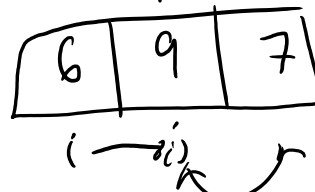
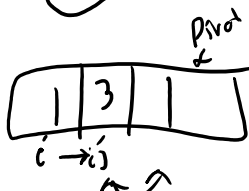
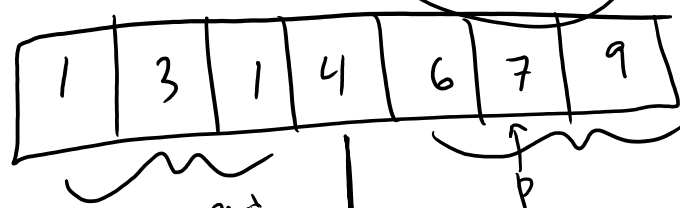
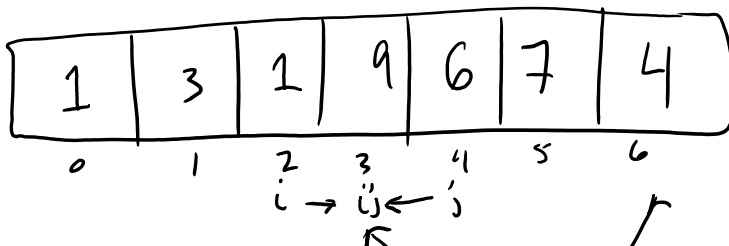
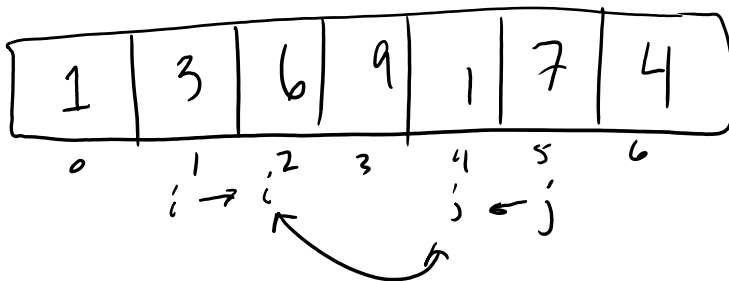
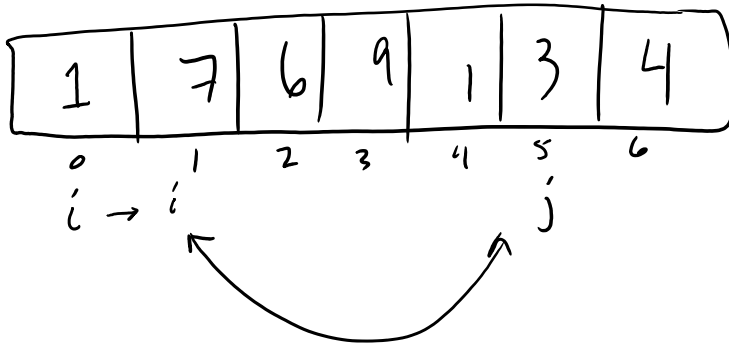
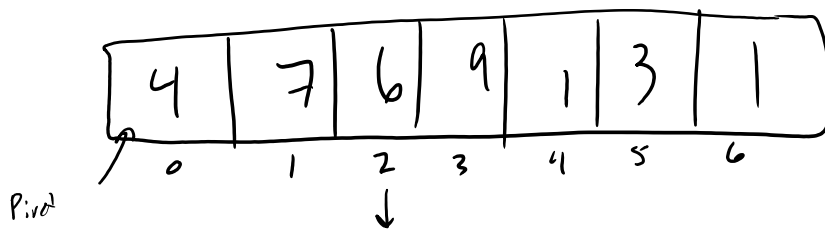
- Step #1: move pivot out of the way (swap with last element)



1. Define $i = \text{front}$; $j = \text{end} - 1$
2. WHILE $\text{numbers}[i] < \text{pivot}$ AND $i < j$
 - a. Increment i
3. WHILE $\text{numbers}[j] > \text{pivot}$ AND $i < j$
 - a. Decrement j
4. If $i \neq j$:
 - a. Swap $\text{numbers}[i]$ with $\text{numbers}[j]$
 - b. Go back to #1
5. Otherwise, loop is done. Swap $\text{numbers}[\text{end}]$ with $\text{numbers}[i]$
6. Repeat process recursively, splitting data on index i



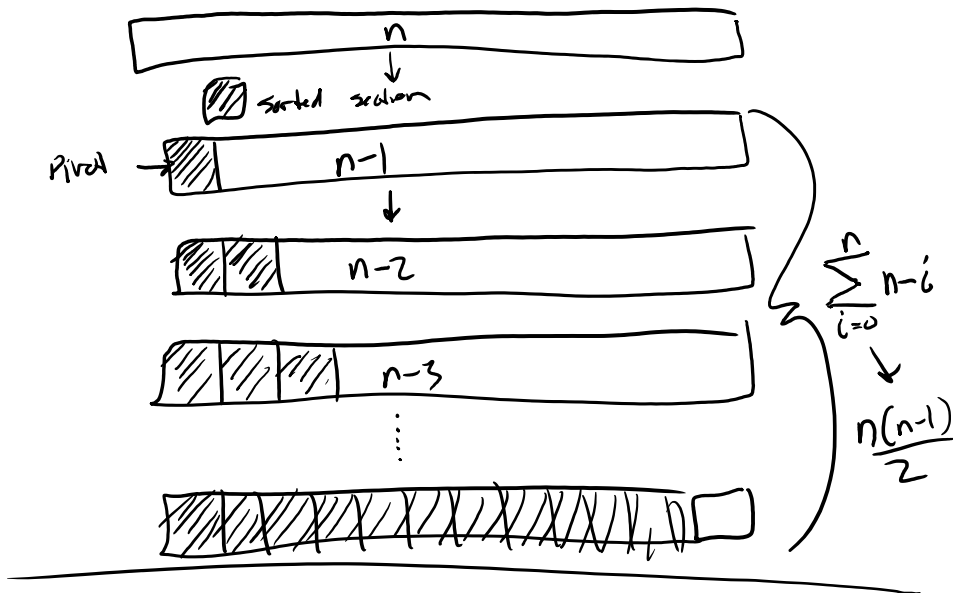
Another example



Analysis of Quicksort

- Consider if we were to always choose the worst possible pivot

- Always pick smallest item in sequence



- Consider a quicksort that always picks the perfect pivot
 - Perfect pivot is the median value

