S.KIMMEL

Announcements

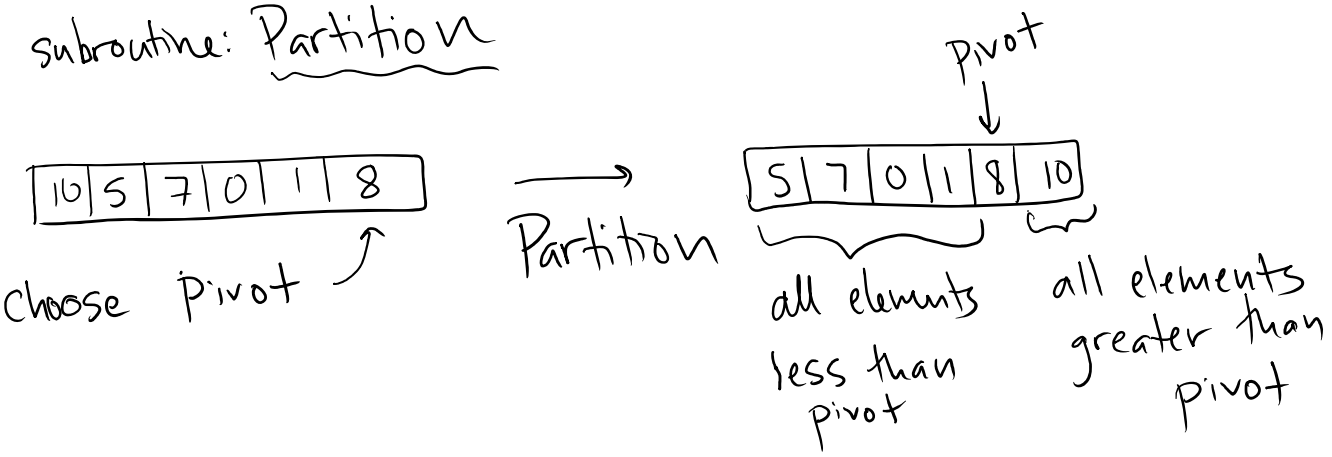
- I am away Wed./Fri
  - Friday: in-class quiz (Prof Andrews)
  - I will post video lectures to website. You may want to come to class and watch discuss together
  - Self-Grade/Reflections: Due by 3 pm Wed to my mailbox or submitted on Canvas
    ↑ outside MBH 633
  - Extra Office Hours on Wed, Thurs, Friday
    Book at Calendly.com/skimmel
    Will use Zoom. I will send Meeting ID
    Go to Middlebury.zoom.us
    Click Join & put in ID
  - * on HW
  - Warner 208, Thurs 3-4:30

S. KIMMEL

Randomization in Recursive Algorithms

QuickSort Review

Key subroutine: Partition



Input: Array A of length n, no repeated elements

Output: Array with sorted elements

QuickSort(array A)

1. Base case
2. pivot = ChoosePivot(A)
3. Partition(A, pivot)
4. Let A_L be array left of pivot, A_R be array right of pivot. } Divide

   ← pivot is not included in recursive calls. Important for our analysis!
5. QuickSort(A_L)
6. QuickSort(A_R) } Conquer

S. KIMMEL

Fix

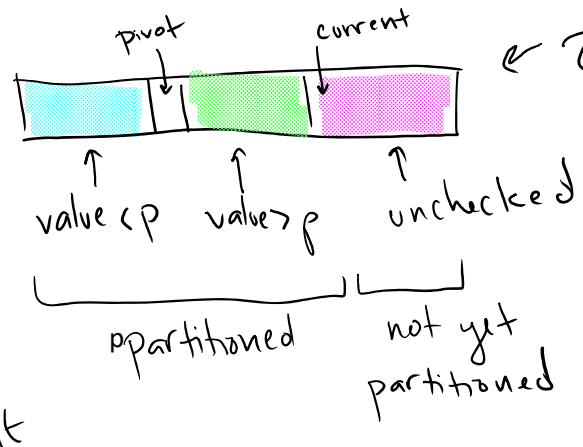
Partition(A, p) p = value of pivot

A: | 10 | 5 | 12 | 0 | 1 | 8 |

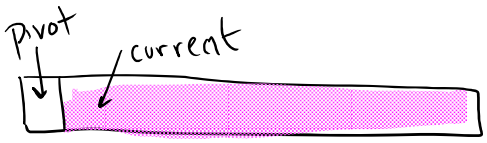
(1) Move pivot to start

| 8 | 5 | 12 | 0 | 1 | 10 |

(2) Loop with loop invariant:



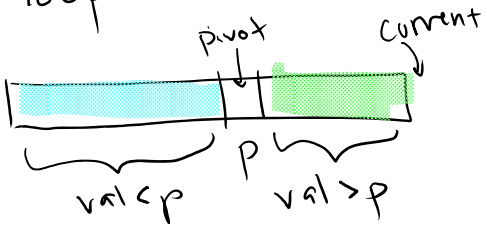
Initially:



At each step of loop:

- compares current to pivot
- does some swaps to maintain invariant including current
- increases current

(3) Maintains loop invariant! So at end



S. KIMMEL

Lemma: Running time of QuickSort is $O(\#\text{ of comparisons})$

Pf: Partition does most of the work, & Partition scaling is $O(\#\text{ of comparisons.})$

Q: How many comparisons are done by Partition on input array of size n?

A: $O(\sqrt{n})$ B: $O(n)$ C: $O(n \log n)$ D: $O(n^2)$

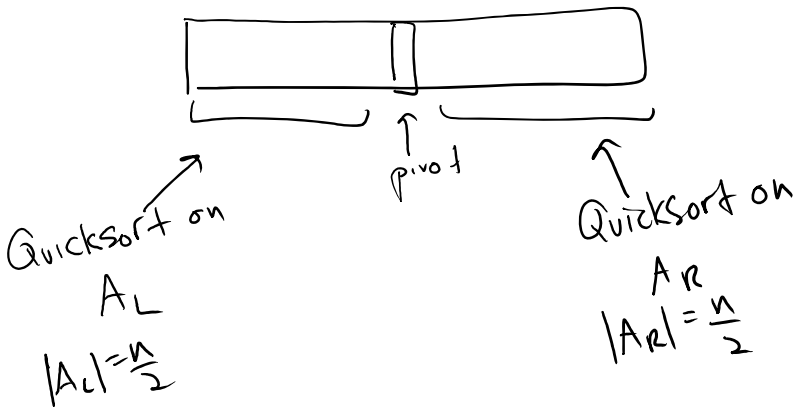
↑ current increases by 1, goes from $1 \to n$.

Q: What is the runtime of QuickSort when the pivot is always chosen to be $\left(\frac{n}{2}\right)^{th}$ largest element of array?

A: $O(\sqrt{n})$ B: $O(n)$ C: $O(n \log n)$ D: $O(n^2)$

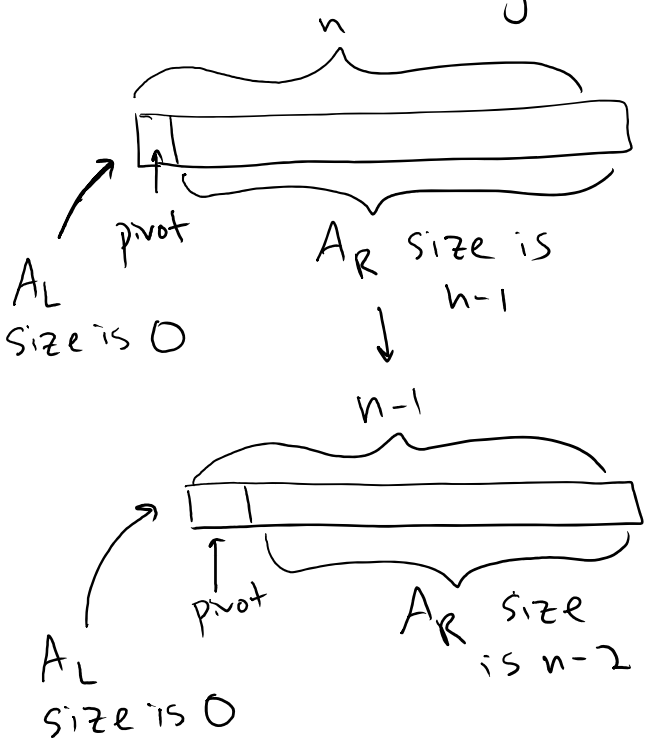
Use master method

$T(n) = 2T\left(\frac{n}{2}\right) + O(n)$



S. KIMMEL

Q: What is the run time of QuickSort if the pivot is always chosen to be the smallest item in array? A: $O(n)$ B: $O(n \log n)$ C: $O(n^{3/2})$ D: $O(n^2)$



Partioning takes time n

" n-1

Run time: $n + n-1 + \dots + 1 = O(n^2)$ Formula: $\frac{n(n-1)}{2}$

Choice of pivot important!

	Bad Choice	Good Choice
Run Time	$O(n^2)$	$O(n \log n)$

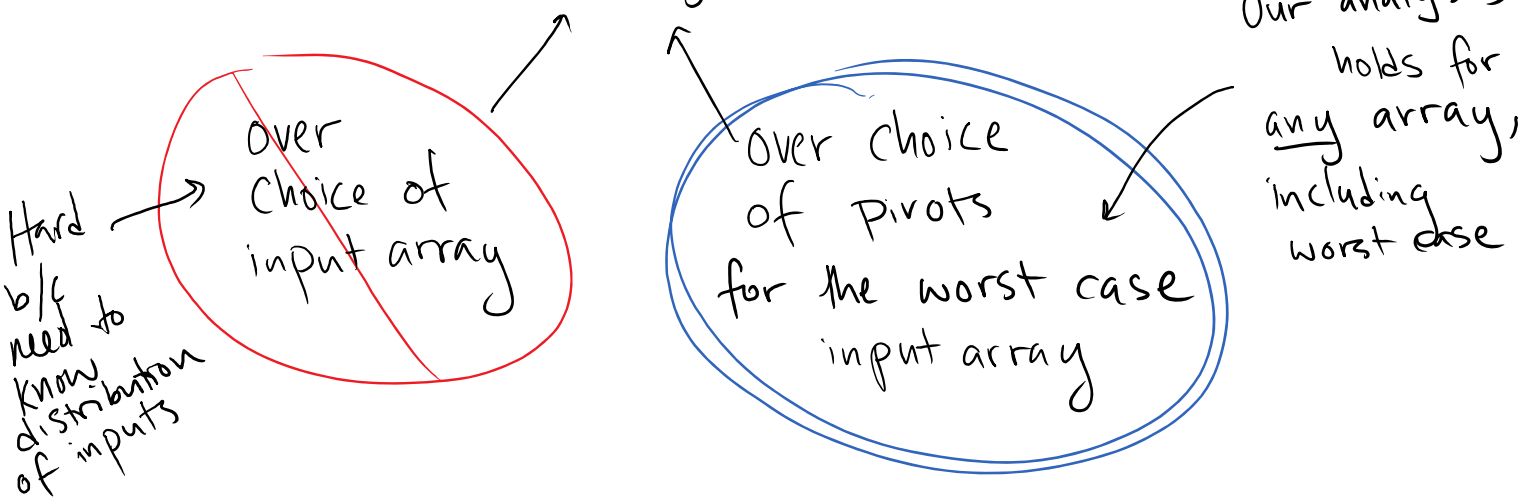
We will show: random choice of pivot is good!

5	7	2	1	11	8	9

pivot = 1, 2, 3, 4, 5, 6, 7 each chosen with probability 1/7

S. KIMMEL

We will calculate average run time:



Why is average run time a good measure?

* The probability of doing asymptotically worse than average is small * (won't prove here.)

S. KIMMEL

Fix array A.

- Sample Space (set of possible events) $S = \{\sigma : \sigma$ is a possible set of pivot choices$\}$

- Probability of $\sigma = Pr(\sigma)$

- Key random variable: $C(\sigma) =$ # of comparison operations required when pivot choices given by σ, for input A