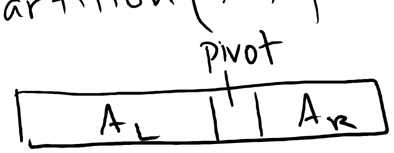
S.KIMMEL

Goals

- Describe impact of pivot choice on runtime of QuickSort
- Evaluate expectation value of random variables

Input: Array A of length n, no repeated elements
Output: Array with sorted elements

QuickSort(array A)

1. If $|A| = 1$: return A
2. pivot = ChoosePivot(A)
3. Partition(A, pivot)
4. 

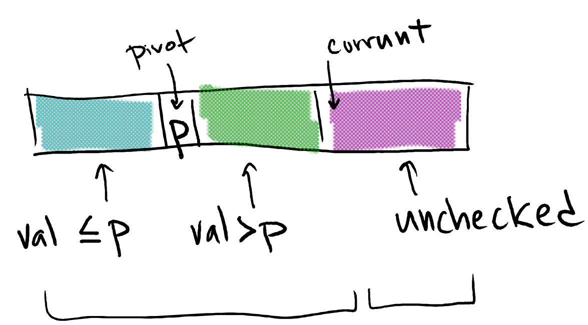
5. QuickSort(A_L)
6. QuickSort(A_R)

Conquer

Partion

(1) Move pivot to front of array

(2) Maintain invariant



S.KIMMEL

Runtime of Quicksort is $O(\#\text{ of comparisons})$

Pf: Partition does most of the work, and runtime of Partition 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: $O(n^2)$

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: $O(n^2)$

Runtime of Quicksort is $O(\#\text{ of comparisons})$

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

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

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

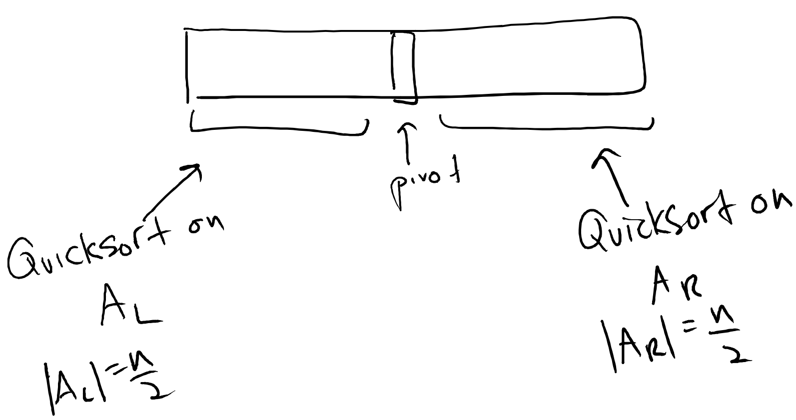
current increases by 1, goes from $1 \to n$.

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: $\boxed{O(n\log n)}$ D: $O(n^2)$

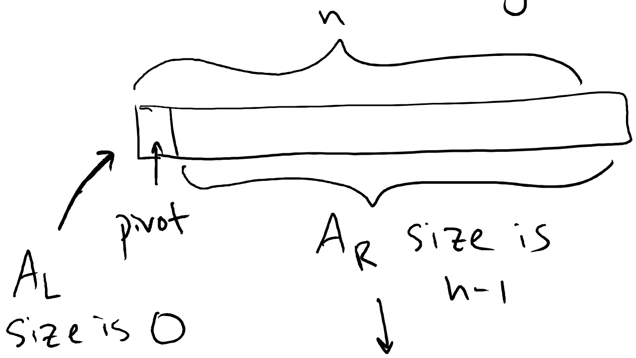
Use master method

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

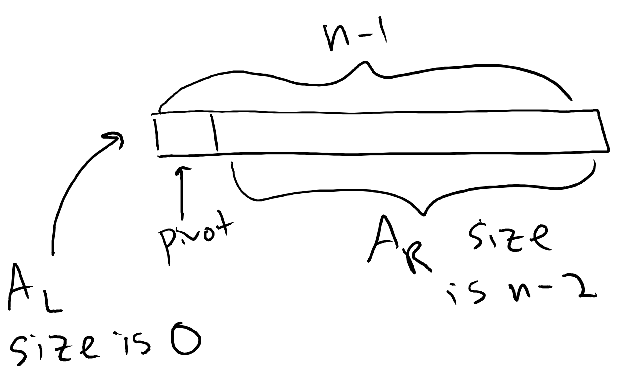


S. KIMMEL

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: $O(n^{3/2})$ $\boxed{D: O(n^2)}$



Partitioning 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: <u>random</u> choice of pivot is good!

S.KIMMEL

Steps to Calculate Average Runtime of Randomized Alg. (Naive)

1. Describe sample space: S of possible random outcomes
2. Calculate probability $p(s)$ for each $s \in S$
3. Create random variable $R: S \to \mathbb{R}$ where $R(s)$ = runtime with outcomes $s \in S$, and evaluate for each $s \in S$
4. Evaluate: $\mathbb{E}[R] = \sum_{s \in S} R(s)p(s)$

1 <u>Sample Space</u> S = set of possible outcomes of random choices

Example:

8	5	7

Q: S = set of possible pivot choices of the algorithm
What is S?

A) $S = \{8, 5, 7\}$

B) $S = \{(8,5,7), (8,7,5), (5,8,7), (5,7,8), (7,5,8), (7,8,5)\}$

C) $S = \{7, (5,7), (5,8), (8,5), (8,7)\}$

S.KIMMEL

- **Sample space** S = set of possible outcomes

Example:

$\boxed{8 \,|\, 5 \,|\, 7}$

S = set of possible pivot choices of the algorithm

A) $S = \{8, 5, 7\}$

B) $S = \{(8,5,7), (8,7,5), (5,8,7), (5,7,8), (7,5,8), (7,8,5)\}$

$\boxed{\text{C)}}$ $S = \{7, (5,7), (5,8), (8,5), (8,7)\}$

If 7 chosen: $\boxed{5 \,|\, 7 \,|\, 8}$ → $\boxed{5}$ $\boxed{8}$ Arrays too small, no partition

If 5 chosen: $\boxed{5 \,|\, 8 \,|\, 7}$ → $\boxed{8 \,|\, 7}$ 8 or 7 can be chosen as pivot

If 8 chosen $\boxed{5 \,|\, 7 \,|\, 8}$ → $\boxed{5 \,|\, 7}$ 5 or 7 can be chosen as pivot

For large arrays... it will be a mess

S.KIMMEL

Step 2: Describe probability

- **Probability** $P: S \to \mathbb{R}$ $P(\sigma)$ = probability of outcome σ occuring

Q. If pivot is chosen randomly at each recursive call of partition, what is $P\left(5,7\right)$?

A) $1/6$

B) $1/5$

C) $1/3$

D) Impossible to determine

S. KIMMEL

- **Probability** $P: S \to \mathbb{R}$ $\quad P(\sigma)$ = probability of outcome σ occuring

Q. If pivot is chosen randomly at each recursive call of partition, what is $P(5,7)$?

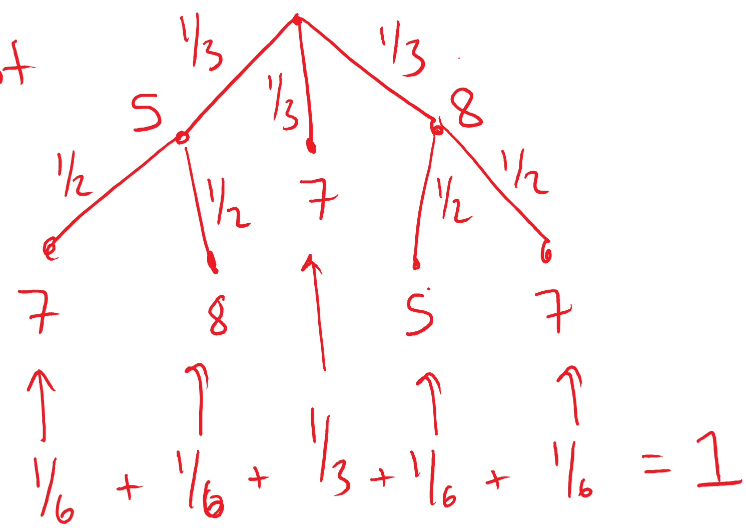
A) $\boxed{1/6}$

B) $1/5$

C) $1/3$

D) Impossible to determine

First Choice of pivot



... For large arrays it is going to be a mess

Calculate $R(s)$ for each $s \in R$ ... makes my head hurt...

S.KIMMEL

Better Approach

1. Describe (generally) the sample space

   ex: $S = \{ s : s$ is a sequence of possible pivot choices for our array$\}$

2. Describe (generally) the run time random variable

   ex: $R(s) =$ # of comparisons over algorithm with pivot choices s.

3. Write $R = \sum_i X_i$ other random variables

   ex: $X_{ij}(s) =$ # of times i^{th} largest element + j^{th} largest element are compared with pivot choices s

   $$R(s) = \sum_{i,j} X_{ij}(s)$$

   To create X_i, think about what are the basic events that are adding an amount of 1 to R

4. $$\mathbb{E}[R] = \sum_i \mathbb{E}[X_i]$$

   linearity of expectation

   $$\mathbb{E}[X_i] = \sum_{s \in S} X_i(s)\, p(s)$$

If choose X_i carefully, this is easier to calculate.

Best X_i = indicator random variable: takes value 0 or 1.