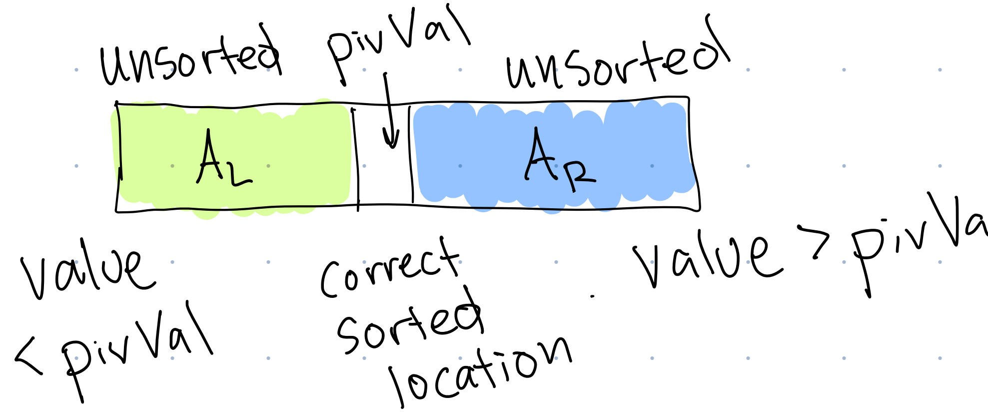
QuickSort

Input: Array A of unique integers
Output: Sorted A

- If $|A| = 1$: Return A
- pivInd ← randomly chosen index, with value pivVal } Preprocessing
- Partition (A, pivInd) } Dividing
- QuickSort (A_L)
- QuickSort (A_R) } Conquering

After Partition:

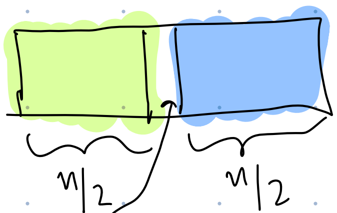


Key Pts

- Partition takes $O(|A|)$ time
- If pivVal is z_i (i^{th} smallest element of A) after Partition, pivVal is at position i.

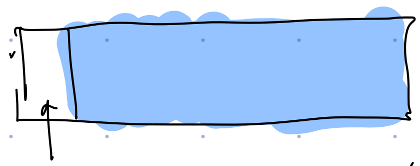
Lucky vs. Unlucky Pivot Choices

Lucky:



pivVal is median array value

Unlucky:



pivVal is smallest/largest array value

1. Suppose you get lucky at every recursive call of QuickSort.

2. Suppose you get unlucky at every recursive call of QuickSort.

- Create recurrence relation for runtime of QuickSort in each case
- Solve recurrence to determine runtime in each case

3. What is Sample Space? Random Variable? Expectation value? Linearity of Expectation?

Lucky vs. Unlucky Pivot Choices

1. Suppose you get <u>lucky</u> at every recursive call or QuickSort.

$$T(n) = \begin{cases} O(1) & \text{if } n \le 1 \\ \underset{\substack{\uparrow \\ A_L}}{T\left(\frac{n}{2}\right)} + \underset{\substack{\uparrow \\ A_R}}{T\left(\frac{n}{2}\right)} + O(n) = 2T\left(\frac{n}{2}\right) + O(n) \end{cases} \overset{\text{Tree formula}}{\Big\}} \underbrace{O(n \log n)}_{\text{Great!}}$$

2. Suppose you get <u>unlucky</u> at every recursive call of QuickSort.

$$T(n) = \begin{cases} O(1) & n \le 1 \\ T(n-1) + O(n) \end{cases} \Bigg\} \Rightarrow \text{Expand + Hope} \quad \underbrace{T(n) = O(n^2)}_{\text{Awful}}$$

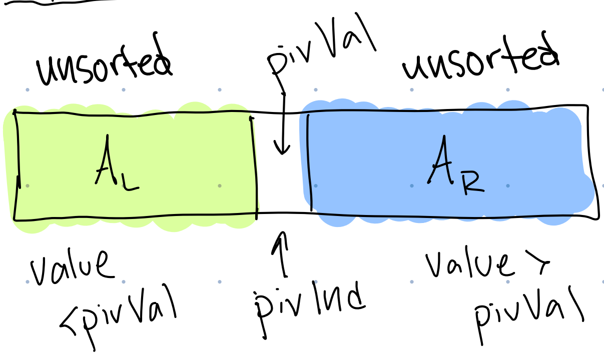
Partition (A, pivInd, pivVal)

- Swap pivot with A[1]
- Current ← 2
- While current ≤ |A|:
  - ☆ If A[current] < pivVal:
    - Swap A[current], pivVal
    - Swap A[pivInd+1], pivVal
  - current ++

← every element of array is compared to pivot

Strategy: count the # of times ☆ is run over the whole algorithm

After Partition:



Analyzing Average Runtime

1. Determine Sample Space S = set of all possible sequences of random events that might occur over the course of alg.

   ex: QuickSort $\rightarrow$ S = set of possible sequences of pivot choices that the alg might make

What is the sample space if QuickSort is run on

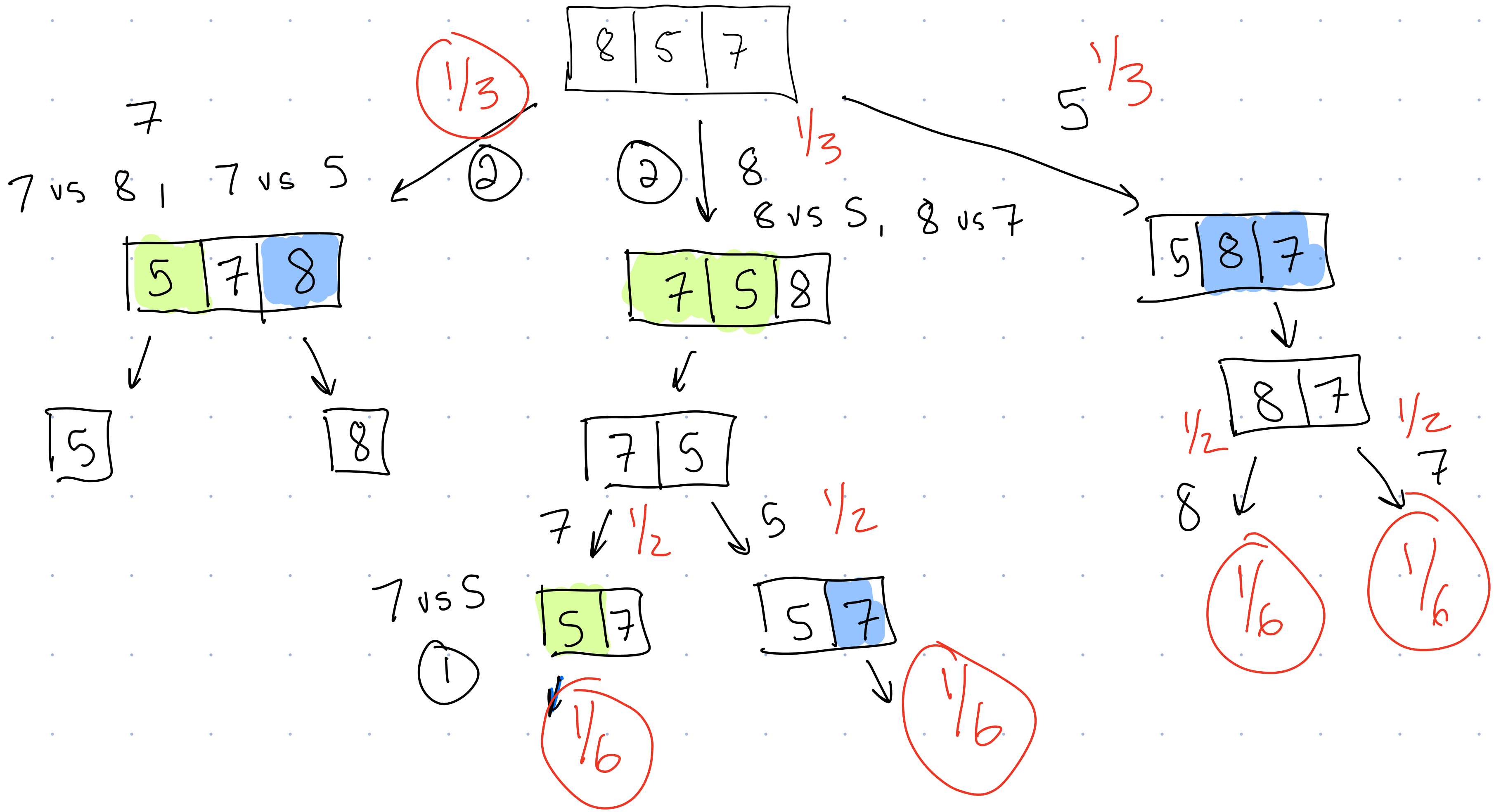
8	5	7

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

B) S = All possible permutations of $\{8, 5, 7\}$

C) S = Power set of $\{8, 5, 7\}$ (set of all subsets of $\{8, 5, 7\}$)

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



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

$R(7) = 2 \qquad R(8,7) = 3$

Analyzing Average Runtime

2. Create a Random Variable that maps each element of the sample space to a number.

$$R : S \to \mathbb{R}$$

\# of times $\star$ is run

ex: QuickSort: $R(\sigma) = \boxed{\#\text{ of comparisons}}$ of 2 elements of our array if pivot sequence σ is chosen.

3. Take Expectation value of R to get average run time

$$\mathbb{E}[R] = \sum_{\sigma \in S} \underbrace{p(\sigma)}_{\text{prob. of } \sigma \text{ occuring}} \cdot R(\sigma)$$

ex: | 8 | 5 | 7 |

(7) (5 7) (5 8) (8 5) (8 7)

$$= \frac{1}{3} \cdot 2 + \frac{1}{6} \cdot 3 + \frac{1}{6} \cdot 3 + \frac{1}{6} \cdot 3 + \frac{1}{6} \cdot 3 = 2\tfrac{2}{3}$$