6s: QuickSort (divide and conquer/randomized)

Goals:
- Recall how to calculate averages
- Start thinking about how to analyze QuickSort average runtime
- QuickSort vs Mergesort

Office Hours: Friday until 1:30. After Break: Wed: 3-4

Calculating Average Runtime

1. Describe sample space S.
   Set of all possible sequence of random outcomes that might occur over the course of the alg.
2. Create random variable: $R : S \to \mathbb{R}$ ← runtime
3. Evaluate $\boxed{\mathbb{E}[R] = \sum_{\sigma \in S} \Pr(\sigma) R(\sigma)}$ $R(\sigma)$ = runtime give random outcomes $\sigma \in S$

Too hard

- Write $R(\sigma) = \sum_i X_i(\sigma)$
- Use linearity of expectation

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

↑ easier

Calculating Average Runtime

1. Describe sample space S.
   Set of all possible sequences of random outcomes that might occur over course of alg. (including in all recursive calls)

pivot choices

What is the sample space if QuickSort is run on the array:

|8|5|7|

S = set of possible sequential pivot choices of the algorithm

A) $S = \{8, 5, 7\}$ ←

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

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

1st pivot: 7 → |5|7|8| A_L A_R $R(7) = 2$

2nd pivot

: 5 |5|8|7| A_R 8 $R(5,8) = 2 + 1 = 3$

7

: 8 |5|7|8| 5

7

2. Create random variable $R : S \to \mathbb{Z}^+$ to track runtime

$R(\sigma)$ = # of comparisons over whole alg (all recursive calls) if σ is sequence of pivot choices

3. Evaluate $\mathbb{E}[R]$ (too hard to do directly)
↓

- Write $R = \sum_i X_i$ ← simpler random variable
- Use linearity of expectation:

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

- evaluate $\mathbb{E}[X_i]$
  then
- evaluate sum

$X_{ij}(\sigma)$ = # of comparisons b/t i^{th} smallest elt and j^{th} smallest elt of A, over whole course of alg, when σ is sequence of pivot choices

z_i is i^{th} smallest elt

ex: |8|5|7|
z_3 z_1 z_2

$X_{23}(\sigma)$ = # of times 7 and 8 are compared

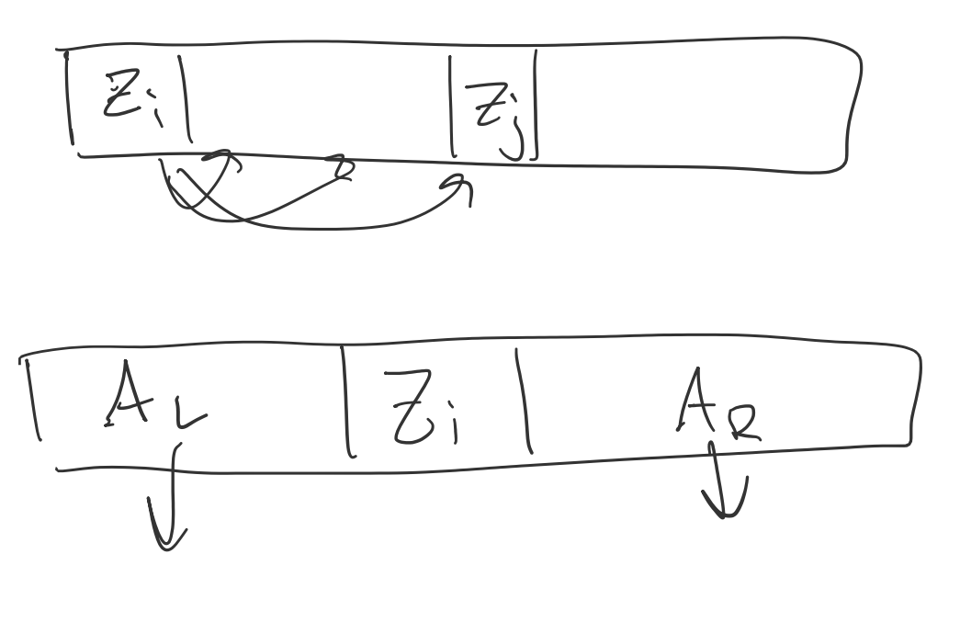
$$R(\sigma) = \sum_{i<j} X_{ij}(\sigma) \qquad \sum_{i=1}^{n-1} \sum_{j=i+1}^{n}$$

Take a moment to reflect on group interaction from last time.
Change in process?

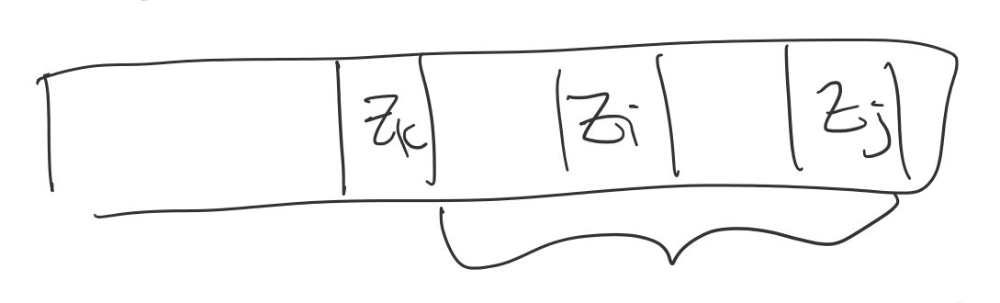
- Suppose z_i, z_j $(i < j)$ are both in an array that is the input to some recursive call of QuickSort. In each of the following cases, when are z_i, z_j compared or not compared in the current recursive call? When are they kept together in the same subarray for the next recursive call or separated? ←
  - z_i or z_j is chosen as pivot. ←
  - z_k is chosen as pivot, $k > i, j$ ←
  - z_k is chosen as pivot, $k < i, j$ ←
  - z_k is chosen as pivot, $i < k < j$ ←
- What values can X_{ij} take? (Hint: only two possible values.) What conditions cause each value?
- What is probability of z_i, z_j being compared?

Story of z_i, z_j:

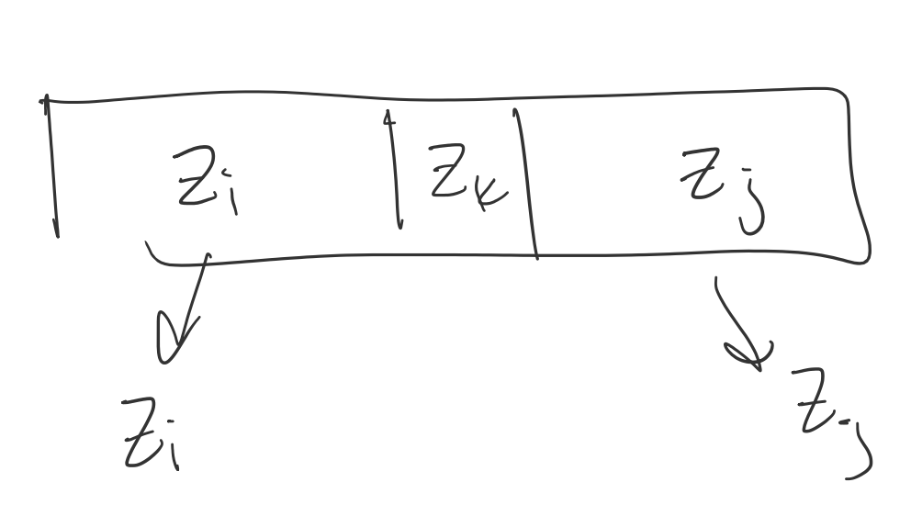
- If z_i or z_j pivot → z_i, z_j compared → never again!



- If pivot $= z_k$, and $k < i, j$ $k > i, j$
  Holding Patterns



- If pivot $= z_k$ $i < k < j$



Not compared
↓
Never compared

$X_{ij} = 0$ or 1