Randomized Quicksort

The Quick Sort Problem

- To sort a given set of numbers
  - In traditional quick sort algorithm, we pick a particular index element as pivot for splitting.
    - Worst Case: $O(n^2)$
    - Average Case: $O(n \log n)$
  - A good pivot can be selected using median finding algorithm but the total complexity will again be $O(n^2)$.
  - So, what if we pick a random element uniformly as pivot and do the partition. We will show that this takes expected $O(n \log n)$ time.

Randomized Quick Sort algorithm

- Assuming all elements are distinct
- We pick a random element x as the pivot and partition the input set S into two sets L and R such:
  - L = numbers less than x
  - R = numbers greater than x
- Recursively sort L and R.
- Return LxR

Analysis of Randomized Quick Sort

- The running time of this algorithm is variable.
- Running time = # of comparisons in this algorithm.
- Expected Running Time, $E[X] = \sum_i x_i * \Pr[X = x_i]$
- Let S be the sorted sequence of the n input numbers.

$S =$

s_1	s_2								s_i					s_j								s_{n-1}	s_n

- Let X_{ij} = 1 if S_i and S_j are compared in the algo
  = 0 otherwise
- Running time = # of comparisons = $\sum_{i=1}^{n} \sum_{j \geq i} X_{ij}$

Analysis cont…

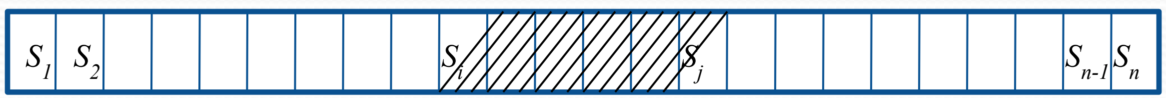
- The expected running time = $E[\sum_{i=1}^{n}\sum_{j\geq i} X_{ij}] = \sum_{i=1}^{n}\sum_{j\geq i} E[X_{ij}]$
- Now, $E[X_{ij}]$ = 1 * Pr[S_i and S_j are compared in our alg.] + 0 * Pr[S_i and S_j are not compared]
- Suppose we have a set of numbers:

  2, 7, 15, 18, 19, 23, 35

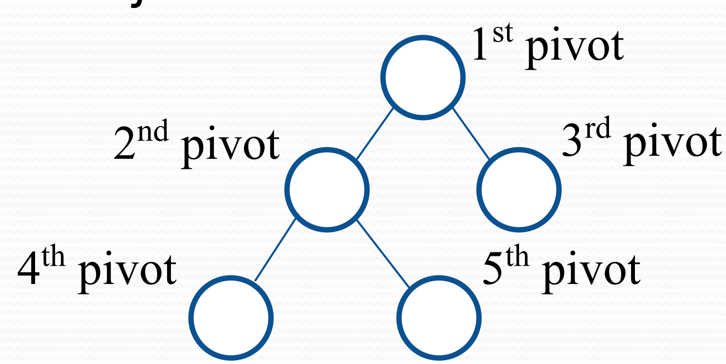
  - In this 18 and 19 will always be compared.
  - 2 and 35 will be compared only if compared at root.

Analysis cont…

- Pr[S_i and S_j are compared in our algo] = Pr[the first element chosen as pivot in set $\{s_i, s_{i+1}, \ldots, s_j\}$ is either s_i or s_j].



- To elements get compared only if they have ancestor relationship in the tree.
- Pr[Picking S_i or S_j] = $\frac{1}{j-i+1} + \frac{1}{j-i+1} = \frac{2}{j-i+1}$



Analysis cont…

- Thus the expected runtime:

$$E[\sum_{i=1}^{n}\sum_{j\geq i}^{n} X_{ij}] = \sum_{i=1}^{n}\sum_{j\geq i}^{n}\frac{2}{j-i+1} \leq \sum_{i=1}^{n}\sum_{j=1}^{n}\frac{2}{j} = O(n\log n)$$

Since, $1+\frac{1}{2}+\frac{1}{3}+\ldots+\frac{1}{n} \approx \ln n$

- This algorithm will always give the right answer though the running time may be different. This is an example of ***Las Vegas algorithms***.