Randomized Quick Sort Algorithm

Lecture Note – 1

Prepared By: Muhammed Miah

01/16/2008

Randomized Quick Sort

- In traditional Quick Sort, we always pick the first element as the pivot for partitioning.
- The worst case runtime is $O(n^2)$ while the expected runtime is $O(n\log n)$ over the set of all input.
- Therefore, some input are born to have long runtime, e.g., an inversely sorted list.

Randomized Quick Sort

- In randomized Quick Sort, we pick randomly an element as the pivot for partitioning.
- The expected runtime of any input is $O(n\log n)$.
- Randomized Quick Sort is a probabilistic algorithm

Randomized Quick Sort (S)

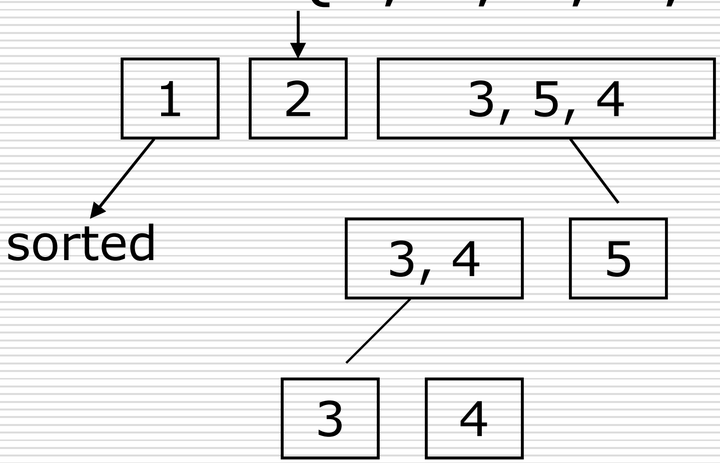
- Input: a set of numbers S,

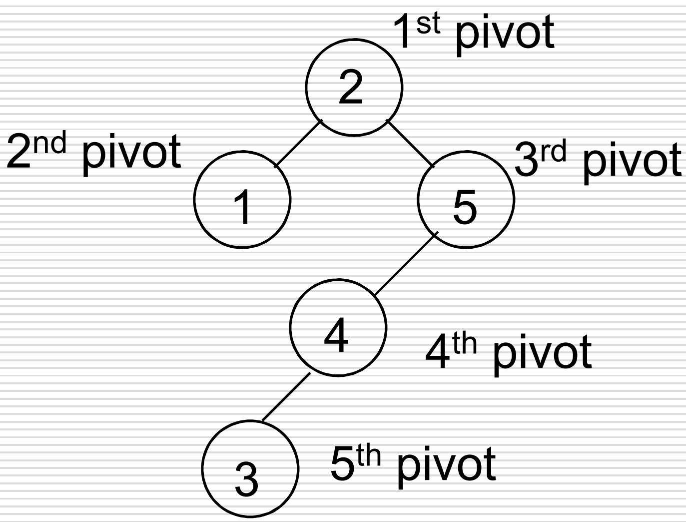
  $S = \{S_1, S_2, S_3, \ldots, S_n\}$

- If |S| > 1 Then,
  1. Select an element X from S at random
  2. Partition S into 3 parts: | L | X | R |
  3. Random Quick Sort (L);
  4. Random Quick Sort (R);

Randomized Quick Sort

- Example:
  - S = {2, 1, 3, 5, 4}





Randomized Quick Sort

- Properties:
  - Running time is unpredictable
  - It has randomization built into it
  - Nothing to do with input order
  - Worst unequal division – when one side has zero number and other side has all numbers
  - Worst case running time n^2

Analysis of "Expected" Running Time of Randomized QS

- Assume, S = {1, 2, 3, 4, …, n)

  Let us consider a Boolean variable, X_{ij} $1 \le i, j \le n, j > i$

  X_{ij} = 1 if it is compared with j

  X_{ij} = 0 otherwise

  # of variables we can create ≈ n^2

	X_{12}	X_{13}	…	$X_{n-1\ n}$	SUM
	1 if X_1 and X_2 are compared, 0 otherwise	…	…	…	SUM(X_{12}, X_{13}, …, $X_{n-1\ n}$)
	…	…	…	…	…
	millions of trees are generated				
AVG.	AVG(X_{12})	AVG(X_{13})	…	…	

Analysis of "Expected" Running Time of Randomized QS

- X_{ij} is a **random** variable

$$E\left[\sum_{i=1}^{n-1}\sum_{j>i}^{n} X_{ij}\right] = \sum_{i=1}^{n-1}\sum_{j>i}^{n} E[X_{ij}]$$

Horizontal definition Vertical definition

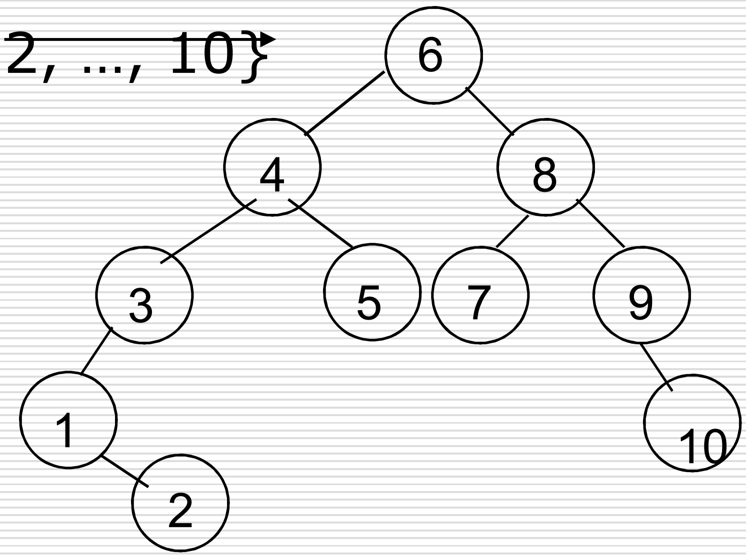
linearity of expectation,

$E[X + Y] = E[X] + E[Y]$

$E[X_{ij}]$ = how often i and j compared

Analysis of "Expected" Running Time of Randomized QS

- Level wise left to right (LR) order:
  consider set of numbers {1, 2, …, 10}
  assume we got the tree:
  level wise order:
  6 4 8 3 5 9 1 10 2



If Min(LR[i], LR[j] > Min(LR of i+1, i+2, …, j-1), then
  i is NOT compared to j
  otherwise it is compared

Analysis of "Expected" Running Time of Randomized QS

$P(X_{ij} = 1) = 2/(j-i+1)$

$E[X_{ij}] = 2/(j-i+1)$

$$\sum_{i=1}^{n-1} \sum_{j>i}^{n} E[X_{ij}] = \sum_{i=1}^{n} \sum_{j>i}^{n} 2/(j-i+1)$$

$$\approx \sum_{i=1}^{n} \sum_{J=i}^{n} 2/(j-i) \approx \sum_{i=1}^{n} \sum_{k=1}^{n-i} 2/k$$

let, j-i = k

$$\leq \sum_{i=1}^{n} \sum_{k=1}^{n} 2/k$$

Analysis of "Expected" Running Time of Randomized QS

$$2\sum_{i=1}^{n}\sum_{k=1}^{n} 1/k = 2n\left(\sum_{k=1}^{n} 1/k\right)$$

$$\sum_{k=1}^{n} 1/k = 1/1 + 1/2 + 1/3 + \ldots + 1/n$$

(Harmonic Series)

$$2n\left(\sum_{k=1}^{n} 1/k\right) = 2n \log n \approx O(n\log n)$$

So, "Expected" running time of randomized quick sort = O(nlogn)