Extra Question Lecture-1 Randomized Find

→ Input: S of integers, k int
→ Output: k^{th} smallest element of S

→ Sorting takes n log n time.
→ Can we do better?

Steps
1. Pick a Random Number y from S.
2. Construct subsets A and B based on S where
   A ≜ Elements of S less than y
   B ≜ Elements of S greater than y
3. a) if |A| = k-1 then return y
   b) if |A| < k-1
      return Find(B, k-(|A|+1))
   c) if |A| > k-1
      return (Find(A, k))

E.g. {2, 8, 3, 9, 7, 16, 4}
We want 5th smallest element.
y = 3
A = {2}
B = {8 9 7 16 4}
Find(B, 5-(1+1))
= Find(B, 3)

Find(B, 3)
y = 9
∴ A = {8, 7, 4}
B = {16}
|A| = 3 > k-1 = 3-1 = 2
∴ Find(A, 3) Recursively (8) is answer.

$T(n,k) \triangleq$ Expected Running Time for input of size n and want to find k^{th} Smallest element

$T(n) \triangleq \max_k T(n,k)$

Size (A)	(B)
0	$n-1$
1	$n-2$
2	$n-3$
⋮	⋮
$n-1$	0
i.e i	$n-i-1$

→ equally likely for each i.e $\frac{1}{n}$ is the probability for each

→ Either you will recurse in (A) or in (B).

→ So, we will consider Max size out of (A), (B).

$$T(n) = (n-1) + \frac{2}{n} * [T(n-1) + T(n-2) + \ldots + T(\tfrac{n}{2}))$$

$$= (n-1) + \frac{2}{n} \sum_{i=n/2}^{n-1} T(i)$$

A $T(n) = (n-1) + \frac{2}{n} \sum_{i=\frac{n}{2}}^{n-1} T(i)$

We will use Guess and Verify Method to solve

Guess: $T(n) \leq 4n$

Verify: $T(n) = (n-1) + \frac{2}{n} \sum_{i=\frac{n}{2}}^{n-1} T(i)$

$$\leq (n-1) + \frac{2}{n} \sum_{i=\frac{n}{2}}^{(n-1)} 4i$$

$$= (n-1) + \frac{8}{n} \sum_{i=\frac{n}{2}}^{n-1} i$$

$$= (n-1) + \frac{8}{n} \left[\sum_{i=1}^{n-1} i - \sum_{i=1}^{n/2-1} i \right]$$

$$= (n-1) + \frac{8}{n} \left[\frac{n(n-1)}{2} - \frac{\left(\frac{n}{2}-1\right) \cdot \frac{n}{2}}{2} \right]$$

$$= (n-1) + 4(n-1) - 2\left(\frac{n}{2}-1\right)$$

$$= n - 1 + 4n - 4 - n + 2$$

$$= 4n - 3$$

$\therefore T(n) \leq 4n-3$ which is Less than $4n$

i.e. $T(n) \leq 4n-3 \leq 4n$

∴ Our guess was Correct and

$T(n) \leq 4n$

$$\therefore \boxed{T(n) = O(n)}$$

∴ This is Las-Vegas algo for finding k^{th} rank element from array in $O(n)$ time.