Created by Aditya Balwani

# Thu 02/11/16

## Find Kth largest element

Claim: There exists atleast 3n/10 elements of A that are smaller than x and 3n/10 elements of A that are larger than X

Proof:

- there are n/10 elements in m that are smaller than x
- For each element of m smaller than x there are 2 more elements of A smaller than x

Recursive call will be a problem of size <= 7n/10

```
(refer to previous notes for this)
findRankKthElement(A, k):
  if(|A| < 5):
    sort(A)
    return A[k]
  1. build m // |m| = n/5
  2. x = findRankKthElement(m, |m|/2)
  3. partition(a, x)
  4. Do a recursive call on one side of the partition, size of partition
<= 7n/10
```

T(n) = running time of findRankKthElement on input of size n

T(1) = 1

T(n) = 9n + T(n/5) + T(7n/10)

T(n) = 5n + 4n + T(2n/10) + T(7n/10)

Lets guess that T(n) <= cn
for some c that we will figure out later

For n<=5, we know that T(n) <= 5n so that statement is true for n<=5 as long as c>=5

So now suppose T(n`) <= cn` for all n` < n then

```
T(n) = 5n + 4n + T(2n/10) + T(7n/10)
  <= 9n + 2cn/10 + 7cn/10
  = (9 + 9c/10)n <= cn

If c >= 90, this is true
```

```
qSort(A):
  x = findRankKthElement(A, |A|/2)
  partition(A, x)
  qSort(A[0, .. , |A|/2])
  qSort(A[|A|/2, .. , |A| - 1])
```

T(1) = 1

T(n) = 94n + 2T(n/2) = 94nlogn

/----------------------------------------------------------/

If 2 input arrays lead to the same leaf of the execution tree then the arrays are equal.

Since there are n! distinct input arrays then there must be >= n! leaves Thus there must exist at least 1 path through the tree with length >= logn!

Stirlings Approximation

n! ~= sqrt(2 * pi * n (n/e)^n)

logn! ~= log(sqrt(2pi)) + (logn)/2 + n(logn-1s) => θ(nlogn)