In Our Inductive Step...

We want to know for what C do we have:

$$k + \frac{C}{5}k + \frac{7C}{10}k \leq Ck$$

$$1 + \frac{C}{5} + \frac{7C}{10} \leq C$$

$$1 \leq \left(1 - \frac{1}{5} - \frac{7}{10}\right) \cdot C$$

$$1 \leq C/10$$

$$10 \leq C$$

What Recurrence Relation does the Running Time Satisfy?

Pseudocode

- **getPivot**(A) returns some pivot for us.
  - How?? We'll see later...
- **Partition**(A,p) splits up A into L, A[p], R.
  - See Lecture 4 IPython notebook for code

Assume getPivot takes time O(n)

- **Select**(A,k):
  - **If** len(A) <= 50:
    - A = **MergeSort**(A)
    - **Return** A[k-1]
  - p = **getPivot**(A)
  - L, pivotVal, R = **Partition**(A,p)
  - **if** len(L) == k-1:
    - return pivotVal
  - **Else if** len(L) > k-1:
    - return **Select**(L, k)
  - **Else if** len(L) < k-1:
    - return **Select**(R, k – len(L) – 1)

Base Case: If the len(A) = O(1), then any sorting algorithm runs in time O(1).

Case 1: We got lucky and found exactly the k'th smallest value!

Case 2: The k'th smallest value is in the first part of the list

Case 3: The k'th smallest value is in the second part of the list

$$T(n) = \begin{cases} T(\text{len}(L)) + O(n) & \text{len}(L) > k-1 \\ T(\text{len}(R)) + O(n) & \text{Len}(L) < k-1 \\ O(n) & \text{len}(L) = k-1 \end{cases}$$

What Recurrence Relation does the Running Time satisfy?

- Lemma says that $|L| \leq \frac{7n}{10} + 5$ and $|R| \leq \frac{7n}{10} + 5$
- Suppose **Partition** runs in time O(n)
- Come up with a recurrence relation for T(n), the running time of **Select**, using the **choosePivot** algorithm we just described.

Pseudocode

- **Select**(A,k):
  - **If** len(A) <= 50:
    - A = **MergeSort**(A)
    - **Return** A[k-1]
  - p = **choosePivot**(A)
  - L, pivotVal, R = **Partition**(A,p)
  - **if** len(L) == k-1:
    - return pivotVal
  - **Else if** len(L) > k-1:
    - return **Select**(L, k)
  - **Else if** len(L) < k-1:
    - return **Select**(R, k – len(L) – 1)

Base Case: If the len(A) = O(1), then any sorting algorithm runs in time O(1).

Case 1: We got lucky and found exactly the k'th smallest value!

Case 2: The k'th smallest value is in the first part of the list

Case 3: The k'th smallest value is in the second part of the list

- CHOOSEPIVOT(A):
  - Split A into m = $\lceil \frac{n}{5} \rceil$ groups, of size <=5 each.
  - **For** i=1, .., m:
    - Find the median within the i'th group, call it p_i
  - p = SELECT([$p_1, p_2, p_3, \ldots, p_m$] , m/2)
  - **return** the index of p in A

This takes time O(1) for each group, since each group has size 5. So that's O(m)=O(n) total in the for loop.

$$T(n) \leq T\left(\frac{7n}{10}\right) + T\left(\frac{n}{5}\right) + O(n)$$