Example: Find the $k^{th}$ Largest Element in Linear Time
(That is, find the item that would be $K^{th}$ is a list sorted from smallest to largest.)

Idea: First let’s consider an algorithm that has poor worst-case time. Choose an arbitrary element $m$ of $L$, partition $L$ into elements $<m$, $=m$, and $>m$, and then recursively look in one partition.

function KLARGEST($k,L$)
  if $k<1$ or $k>|L|$ then ERROR ($k$ is out of range)
  else begin
    • Let $m$ be any element of $L$ (e.g., pick $m$ at random from $L$).
    • Form the lists $A$, $B$, and $C$ of the elements in $L$ that are $<m$, $=m$, and $>m$.
    • if $k \leq |A|$ then return KLARGEST($k,A$)
      else if $k \leq (|A|+|B|)$ then return $m$
      else return KLARGEST($(k-|A|-|B|),C$)
  end
end

(Note: $|L|$ denotes the number of items in $L$.)
Idea: Find an $m$ which is close to the median of $L$, partition $L$ into elements $<m$, $=m$, and $>m$, and then recursively look in one partition.

function KLARGEST($k$, $L$)
  if $k < 1$ or $k > |L|$ then ERROR ($k$ is out of range)
  else if $|L| < 20$ then sort $L$ and return the $k^{th}$ element
  else begin
    1. Divide $L$ into $\left\lfloor \frac{n}{5} \right\rfloor$ "mini-lists" of 5 elements each, and at most 4 leftovers.
    2. $M :=$ a list of the medians of the mini-lists
    3. $m :=$ KLARGEST($\left\lfloor \frac{|M|}{2} \right\rfloor$, $M$)
    4. Form the lists $A$, $B$, and $C$ of the elements in $L$ that are $<m$, $=m$, and $>m$.
    5. if $k \leq |A|$ then return KLARGEST($k$, $A$)
      else if $k \leq (|A| + |B|)$ then return $m$
      else return KLARGEST($(|k| - |A| - |B|)$, $C$)
  end
end
Lists $A$ and $C$ must have $<3/4n$ elements:

- Divide and conquer algorithms typically work best when sub-problems have equal size.
- Although $m$ may not be the median of $L$, neither $A$ nor $C$ gets too large a fraction of $L$.
- Since at least $\lceil |M|/2 \rceil$ of the mini-lists contain a median that is $\leq m$ and at least two other elements that are $\leq m$, at least $\lceil 3|M|/2 \rceil \geq \lceil 3n/10 \rceil$ of the elements of $L$ are $\leq m$, and so $|C| \leq \lceil 7n/10 \rceil$.
- Similarly, since at least $|M| - \lceil |M|/2 \rceil + 1 = \lceil |M|/2 \rceil + 1 \geq \lceil |M|/2 \rceil$ of the mini-lists contain a median that is $\geq m$ and at least two others that are $\geq m$, it follows that $|A| \leq \lceil 7n/10 \rceil$.
- Hence, both $|A|$ and $|C|$ are $\leq (3/4)n$, since $n \geq 20$ implies:
  \[ \lceil 7n/10 \rceil \leq (7n/10)+1 \leq (7n/10) + 0.05n = 0.75n = (3/4)n \]
Time and space used by KLARGEST:

- The space is $O(n)$.
- There is a $c$ such that $T(n) \leq cn$ if $n<20$, and for $n>20$, since there is one recursive call in Step 3 on a list of size $\leq (n/5)$, one in Step 5 on a list of size $\leq (3/4)n$, and everything else is $O(n)$:
  
  $$T(n) \leq T(.2n) + T(.75n) + O(n)$$

  A proof by induction now shows that $T(n) \leq 20cn$. For $n<20$, $cn<20cn$, and for $n \geq 20$, by applying the inductive hypothesis, $T(n) \leq 4cn + 15cn + cn = 20cn$.

Practical considerations: Intuitively, the time is linear because $(1/5) + (3/4) < 1$. However, because the inequality is close, the constant is poor (choosing $m$ randomly may give a more practical expected time algorithm).