order statistics

Blum, Floyd, Pratt, Rivest, Tarjan 1972

finding i'th smallest element out of n in linear time

Algorithm 15 Select

Input: array $a[1..n]$, integer i

Output: the ith largest element in a

1: If $n \leq 60$, then find the ith largest element by sorting.

2: Divide the n elements in $\lfloor n/5 \rfloor$ groups of 5 elements. At most 4 elements remain.

3: Find the median of each of the $\lfloor n/5 \rfloor$ groups by sorting.

4: Recursively call Select to find the median m of the $\lfloor n/5 \rfloor$ medians. (Once we found m, we forget about the groups.)

5: Use the procedure Partition on a with m as the pivot element.

6: Let q be the position of m in the array.

7: If $q = i$, then return m.

8: If $i < q$, then call Select$(a[1..q-1], i)$. Else call Select$(a[q+1..n], i-q)$.

finding i'th smallest element out of n in linear time

Algorithm 15 Select

Input: array $a[1..n]$, integer i

Output: the ith largest element in a

1: If $n \leq 60$, then find the ith largest element by sorting.

2: Divide the n elements in $\lfloor n/5 \rfloor$ groups of 5 elements. At most 4 elements remain.

3: Find the median of each of the $\lfloor n/5 \rfloor$ groups by sorting.

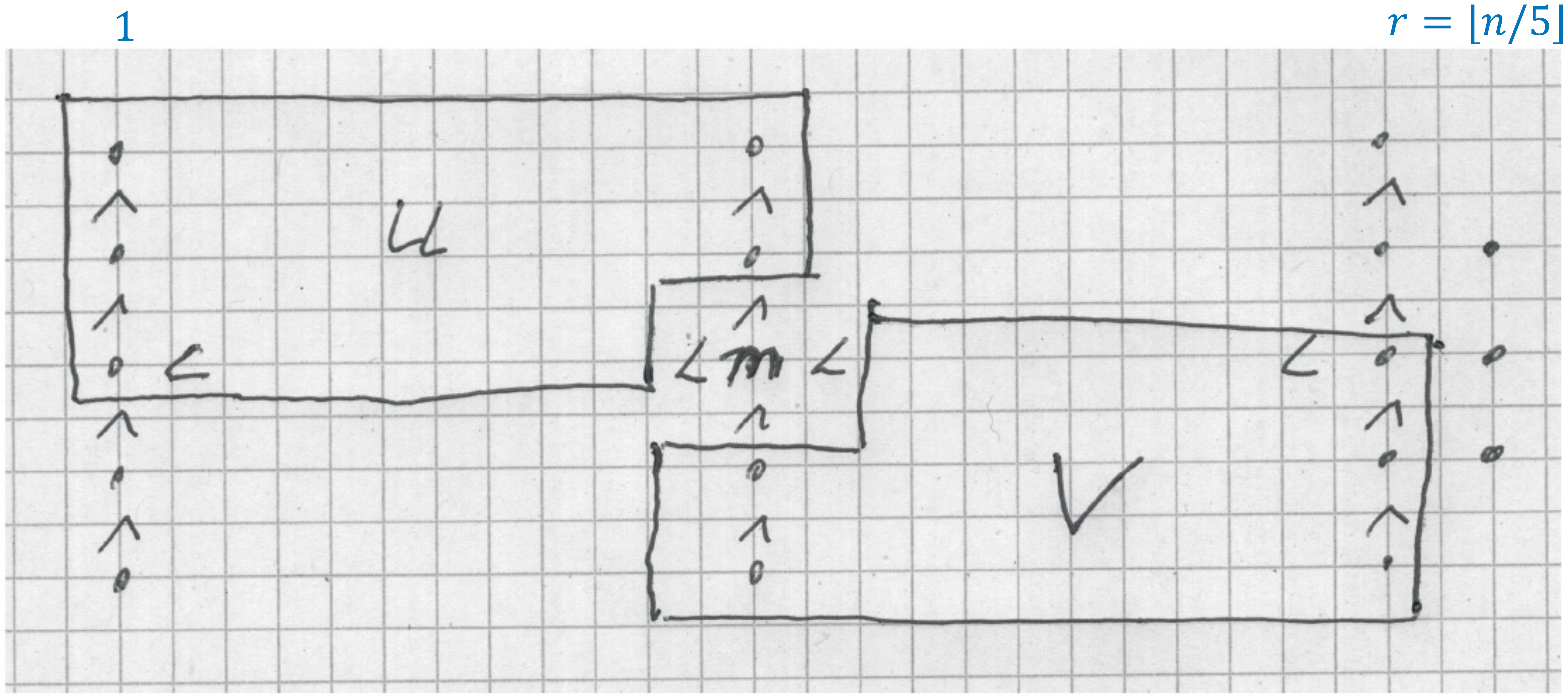
4: Recursively call Select to find the median m of the $\lfloor n/5 \rfloor$ medians. (Once we found m, we forget about the groups.)

5: Use the procedure Partition on a with m as the pivot element.

6: Let q be the position of m in the array.

7: If $q = i$, then return m.

8: If $i < q$, then call Select$(a[1..q-1], i)$. Else call Select$(a[q+1..n], i-q)$.



finding i'th smallest element out of n in linear time

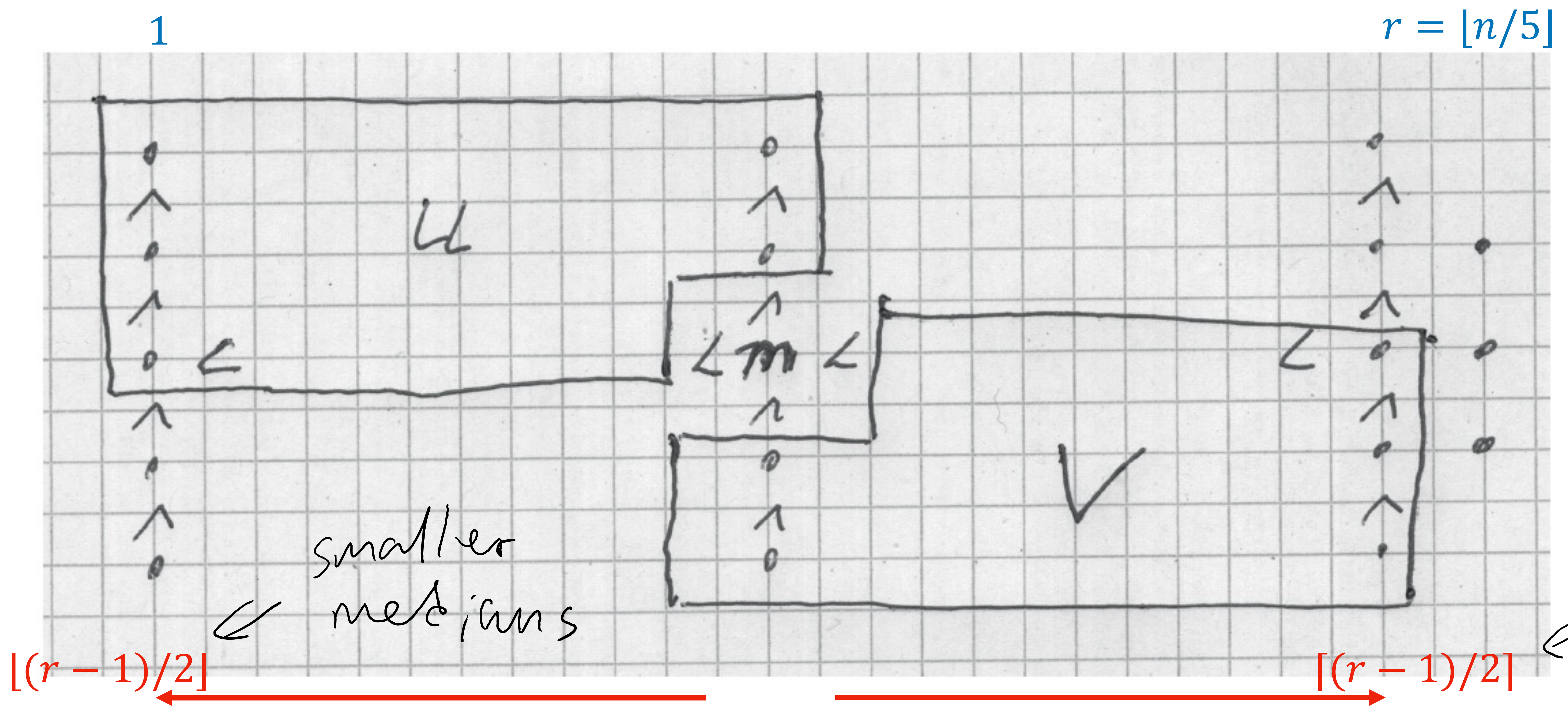
Algorithm 15 Select

Input: array $a[1..n]$, integer i

Output: the ith largest element in a

1: If $n \leq 60$, then find the ith largest element by sorting.
2: Divide the n elements in $\lfloor n/5 \rfloor$ groups of 5 elements. At most 4 elements remain.
3: Find the median of each of the $\lfloor n/5 \rfloor$ groups by sorting.
4: Recursively call Select to find the median m of the $\lfloor n/5 \rfloor$ medians. (Once we found m, we forget about the groups.)
5: Use the procedure Partition on a with m as the pivot element.
6: Let q be the position of m in the array.
7: If $q = i$, then return m.
8: If $i < q$, then call Select$(a[1..q-1], i)$. Else call Select$(a[q+1..n], i-q)$.

$r =$ number of groups with 5 elements $= \lfloor \frac{n}{5} \rfloor$



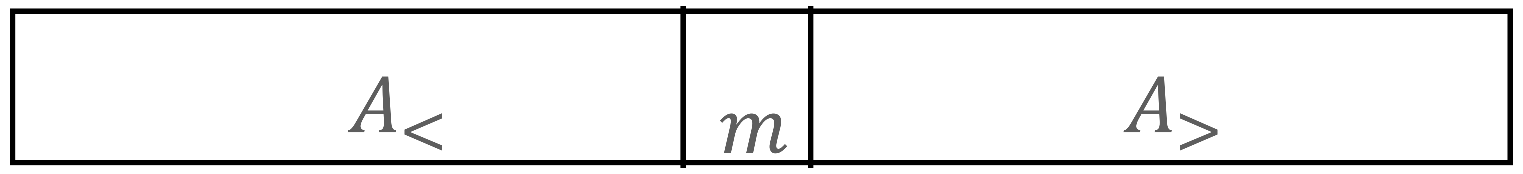
larger medians

finding i'th smallest element out of n in linear time

Algorithm 15 Select

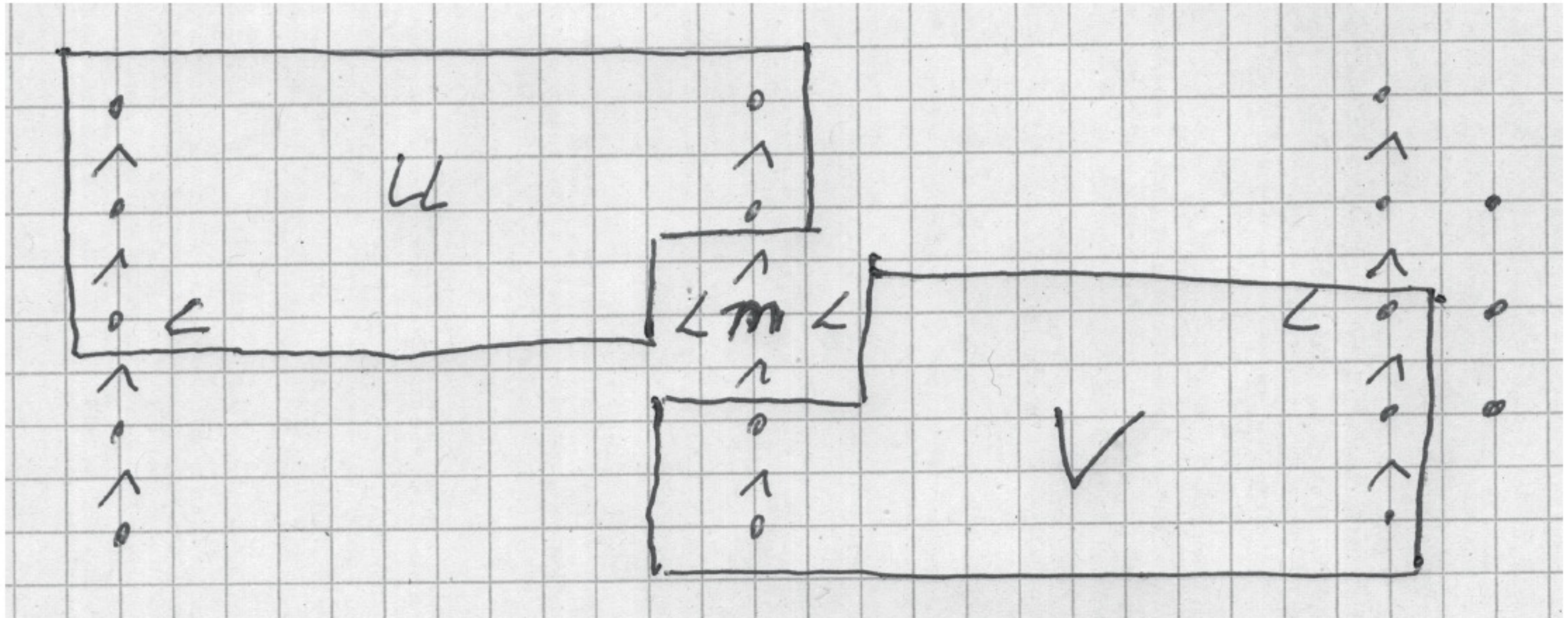
Input: array $a[1..n]$, integer i
Output: the ith largest element in a

1: If $n \leq 60$, then find the ith largest element by sorting.
2: Divide the n elements in $\lfloor n/5 \rfloor$ groups of 5 elements. At most 4 elements remain.
3: Find the median of each of the $\lfloor n/5 \rfloor$ groups by sorting.
4: Recursively call Select to find the median m of the $\lfloor n/5 \rfloor$ medians. (Once we found m, we forget about the groups.)
5: Use the procedure Partition on a with m as the pivot element.
6: Let q be the position of m in the array.
7: If $q = i$, then return m.
8: If $i < q$, then call Select$(a[1..q-1], i)$. Else call Select$(a[q+1..n], i-q)$.



$$
\begin{aligned}
A_< &= \{a(i) \mid a(i) < m\} \\
A_> &= \{a(i) \mid a(i) > m\} \\
q &= \#A_< + 1
\end{aligned}
$$

1 $\qquad$ $r = \lfloor n/5 \rfloor$



$\lfloor (r-1)/2 \rfloor$ $\qquad$ $\lceil (r-1)/2 \rceil$

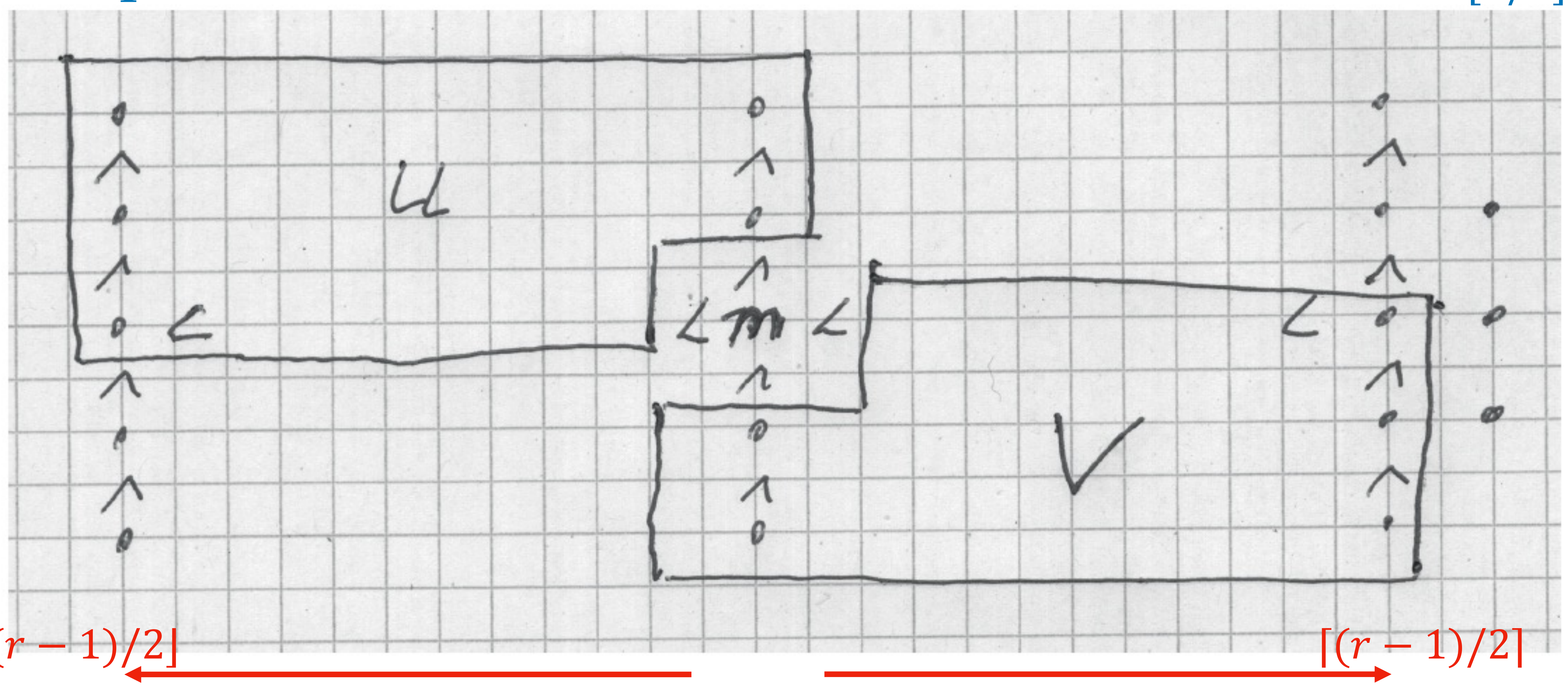
finding i'th smallest element out of n in linear time

Algorithm 15 Select

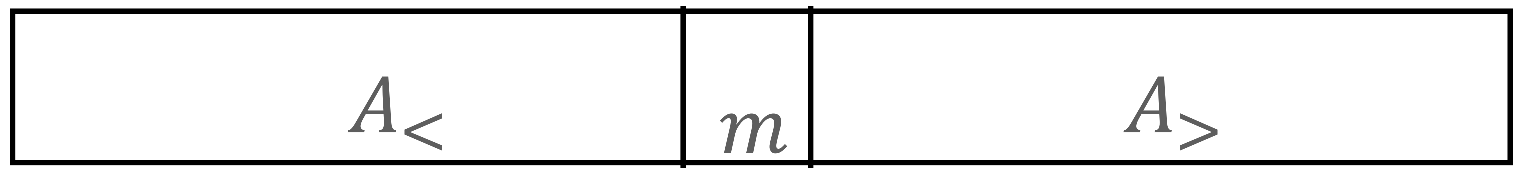
Input: array $a[1..n]$, integer i
Output: the ith largest element in a
1: If $n \leq 60$, then find the ith largest element by sorting.
2: Divide the n elements in $\lfloor n/5 \rfloor$ groups of 5 elements. At most 4 elements remain.
3: Find the median of each of the $\lfloor n/5 \rfloor$ groups by sorting.
4: Recursively call Select to find the median m of the $\lfloor n/5 \rfloor$ medians. (Once we found m, we forget about the groups.)
5: Use the procedure Partition on a with m as the pivot element.
6: Let q be the position of m in the array.
7: If $q = i$, then return m.
8: If $i < q$, then call Select$(a[1..q-1], i)$. Else call Select$(a[q+1..n], i-q)$.

---

1 $\qquad$ $r = \lfloor n/5 \rfloor$



$\lfloor (r-1)/2 \rfloor$ $\qquad$ $\lfloor (r-1)/2 \rfloor$



$$\begin{aligned} A_< &= \{a(i) \mid a(i) < m\} \\ A_> &= \{a(i) \mid a(i) > m\} \\ q &= \#A_< + 1 \end{aligned}$$

Lemma 1.

$$q - 1 \geq \frac{3n}{10} - 3 \,,\; n - q \geq \frac{3n}{10} - 3$$

Proof.

$$q - 1 \geq \#U \,,\; n - q \geq \#V \geq \#U$$

$$\left\lfloor \frac{n}{5} \right\rfloor > \frac{n}{5} - \frac{4}{5}$$

$$\begin{aligned} \#U &= \left\lfloor \frac{r-1}{2} \right\rfloor \cdot 3 + 2 \\ &\geq \frac{r-2}{2} \cdot 3 + 2 \\ &\geq \frac{3}{2}\left(\frac{n}{5} - \frac{4}{5} - 2\right) + 2 \\ &= \frac{3n}{10} - \frac{12}{10} - 3 + 2 \\ &= \frac{3n}{10} - 2.2 \end{aligned}$$

finding i'th smallest element out of n in linear time

Algorithm 15 Select

Input: array $a[1..n]$, integer i

Output: the ith largest element in a

1: If $n \leq 60$, then find the ith largest element by sorting.
2: Divide the n elements in $\lfloor n/5 \rfloor$ groups of 5 elements. At most 4 elements remain.
3: Find the median of each of the $\lfloor n/5 \rfloor$ groups by sorting.
4: Recursively call Select to find the median m of the $\lfloor n/5 \rfloor$ medians. (Once we found m, we forget about the groups.)
5: Use the procedure Partition on a with m as the pivot element.
6: Let q be the position of m in the array.
7: If $q = i$, then return m.
8: If $i < q$, then call Select$(a[1..q-1], i)$. Else call Select$(a[q+1..n], i-q)$.

$$\boxed{t(n) \leq \begin{cases} t(\lfloor n/5 \rfloor + t(\lceil 7n/10 \rceil + 2) + d \cdot n & n > 60 \\ n \cdot \log 60 & n \leq 60 \end{cases}}$$

$$n - \left(\frac{3n}{10} - 3\right) = \frac{7n}{10} n + 3$$

subtract one more for m

$A_<$	m	$A_>$

$$\begin{aligned} A_< &= \{a(i) \mid a(i) < m\} \\ A_> &= \{a(i) \mid a(i) > m\} \\ q &= \#A_< + 1 \end{aligned}$$

Lemma 1.

$$q - 1 \geq \frac{3n}{10} - 3 \text{ , } n - q \geq \frac{3n}{10} - 3$$

Proof.

$$q - 1 \geq \#U \text{ , } n - q \geq \#V \geq \#U$$

$$\begin{aligned} \#U &= \lfloor \frac{r-1}{2} \rfloor \cdot 3 + 2 \\ &\geq \frac{r-2}{2} \cdot 3 + 2 \\ &\geq \frac{3}{2}\left(\frac{n}{5} - \frac{4}{5} - 2\right) + 2 \\ &= \frac{3n}{10} - \frac{12}{10} - 3 + 2 \\ &= \frac{3n}{10} - 2.2 \end{aligned}$$

finding i'th smallest element out of n in linear time

Algorithm 15 Select

Input: array $a[1..n]$, integer i

Output: the ith largest element in a

1: If $n \leq 60$, then find the ith largest element by sorting.
2: Divide the n elements in $\lfloor n/5 \rfloor$ groups of 5 elements. At most 4 elements remain.
3: Find the median of each of the $\lfloor n/5 \rfloor$ groups by sorting.
4: Recursively call Select to find the median m of the $\lfloor n/5 \rfloor$ medians. (Once we found m, we forget about the groups.)
5: Use the procedure Partition on a with m as the pivot element.
6: Let q be the position of m in the array.
7: If $q = i$, then return m.
8: If $i < q$, then call Select$(a[1..q-1], i)$. Else call Select$(a[q+1..n], i-q)$.

$$\boxed{t(n) \leq \begin{cases} t(\lfloor n/5 \rfloor + t(\lceil 7n/10 \rceil + 2) + d \cdot n & n > 60 \\ n \cdot \log 60 & n \leq 60 \end{cases}}$$

Try induction step and induction start with $t(n) \leq cn$

$$\begin{aligned}
t(n) &\leq c\lfloor n/5 \rfloor + c(\lceil 7n/10 \rceil + 2) + d \cdot n \\
&\leq nc(1/5 + 7/10) + 3c + d \cdot n \\
&= nc(9/10) + 3c + d \cdot n \\
&\leq cn \\
3c + d \cdot n &\leq cn/10 \\
3c/n + d &\leq c/10 \\
c/20 + d &\leq c/10 \quad (n > 60) \\
\log 60 \leq 20 \cdot d &\leq c \quad \text{(for start of induction)}
\end{aligned}$$