

Algorithm: Sel(numbers,k)

Given: numbers x_1, \dots, x_n , k, output k'th smallest number.

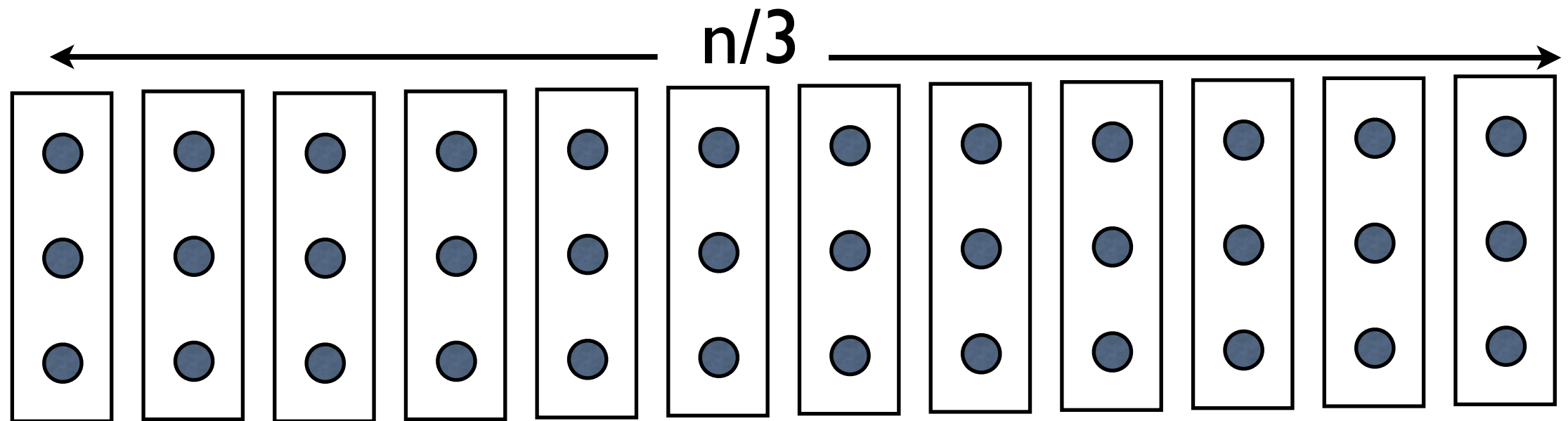
Algorithm: Sel(numbers,k)

Given: numbers x_1, \dots, x_n , k, output k'th smallest number.

Sort the numbers! $O(n \log n)$ time.

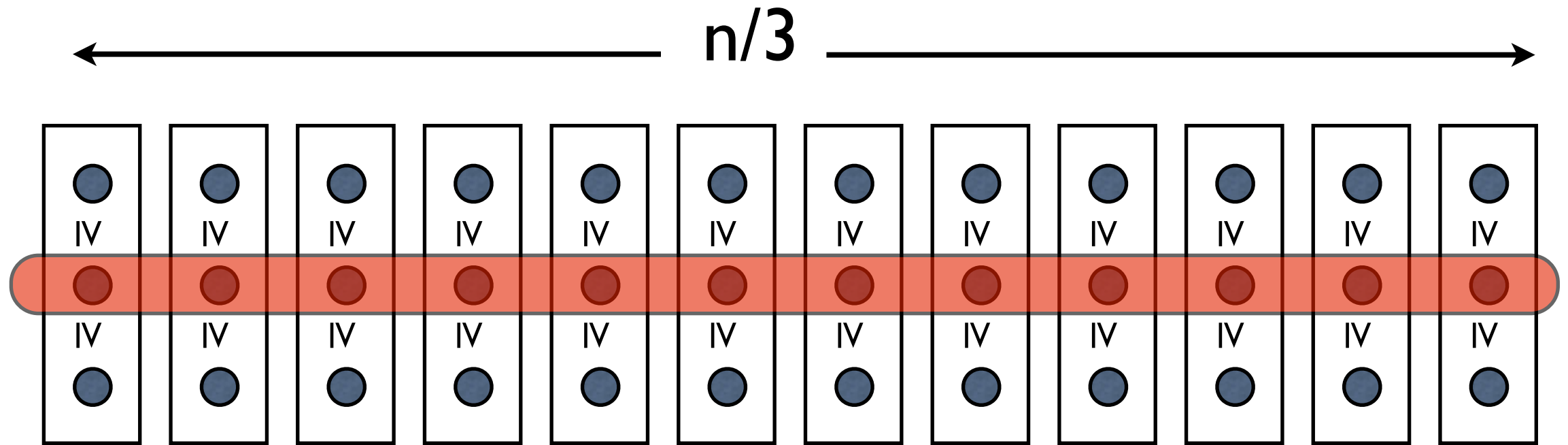
Can we do better?

Given: x_1, \dots, x_n , k , output k 'th smallest number.



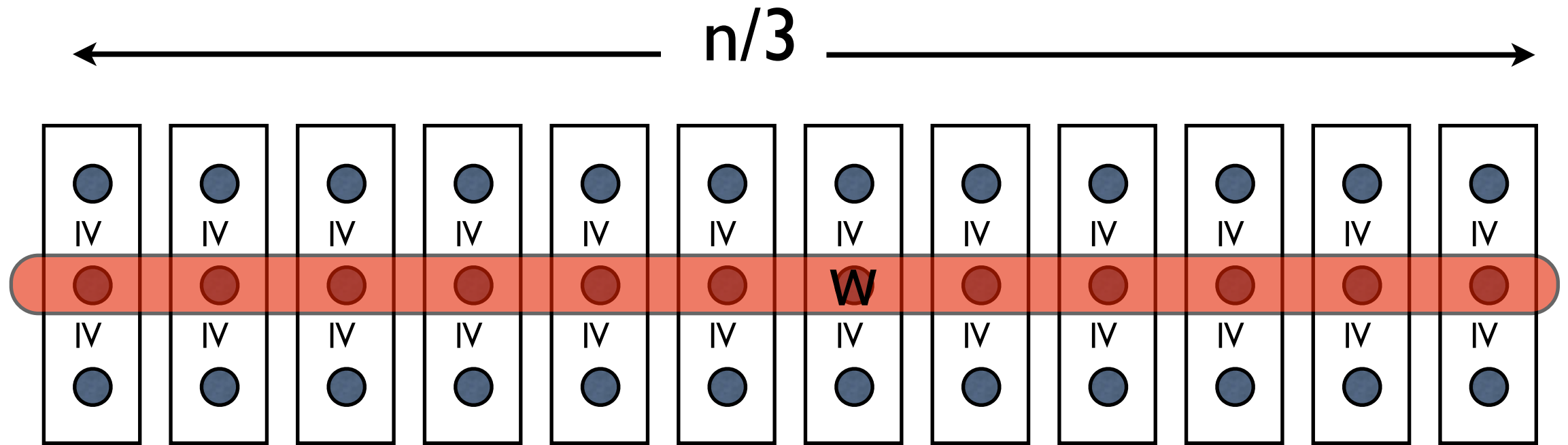
1. Partition numbers into sets of size 3

Algorithm: Sel(numbers, k)



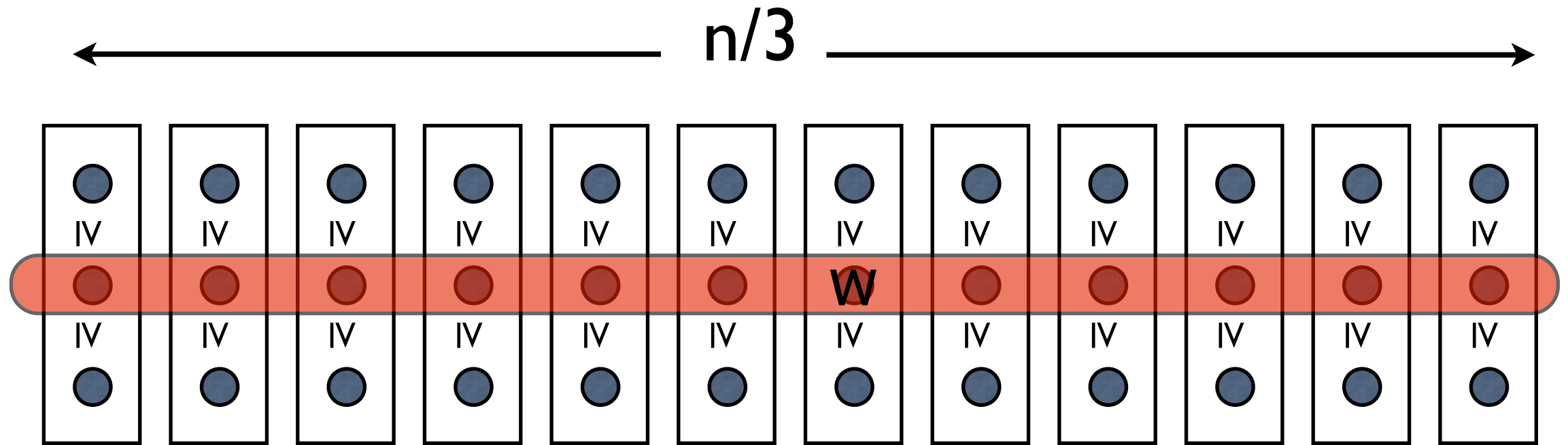
1. Partition numbers into sets of size 3
2. Sort each set

Algorithm: Sel(numbers,k)



Algorithm: Sel(numbers,k)

1. Partition numbers into sets of size 3
2. Sort each set
3. $w = \text{Sel}(\text{red oval}, n/6)$

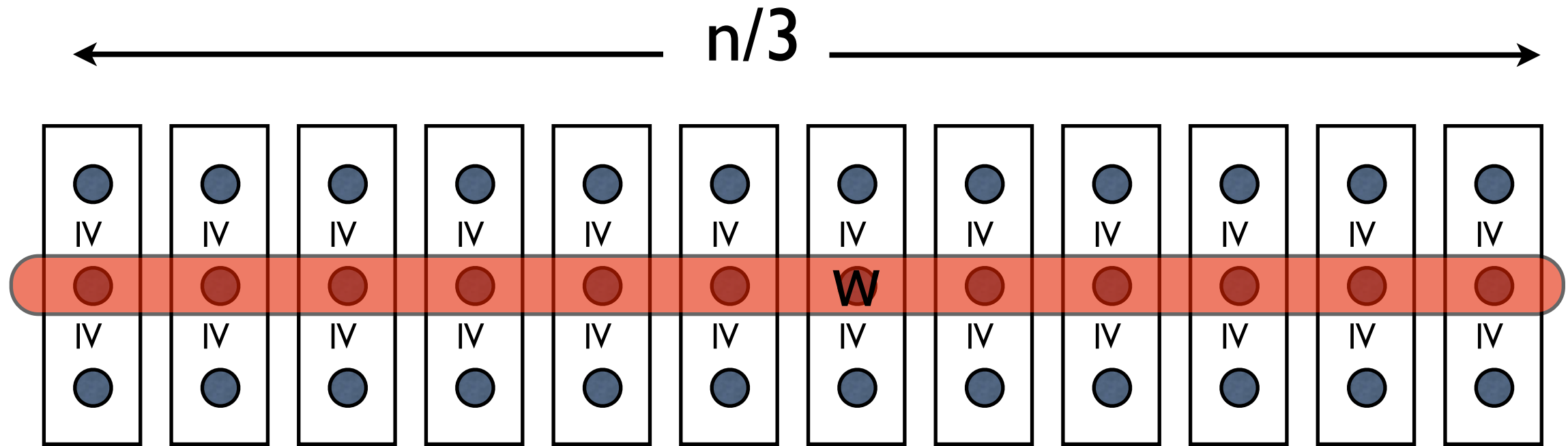


1. Partition numbers into sets of size 3
2. Sort each set
3. $w = \text{Sel}(\text{red bar}, n/6)$

$$\begin{aligned}
 S_L(w) &= \{x_i \mid x_i < w\} \\
 S_E(w) &= \{x_i \mid x_i = w\} \\
 S_G(w) &= \{x_i \mid x_i > w\}
 \end{aligned}
 \left. \vphantom{\begin{aligned} S_L(w) \\ S_E(w) \\ S_G(w) \end{aligned}} \right\}$$

Can be computed in linear time

Algorithm: Sel(numbers,k)



Algorithm: Sel(numbers,k)

1. Partition numbers into sets of size 3
2. Sort each set
3. $w = \text{Sel}(\text{red oval}, n/6)$

$$S_L(w) = \{x_i \mid x_i < w\}$$

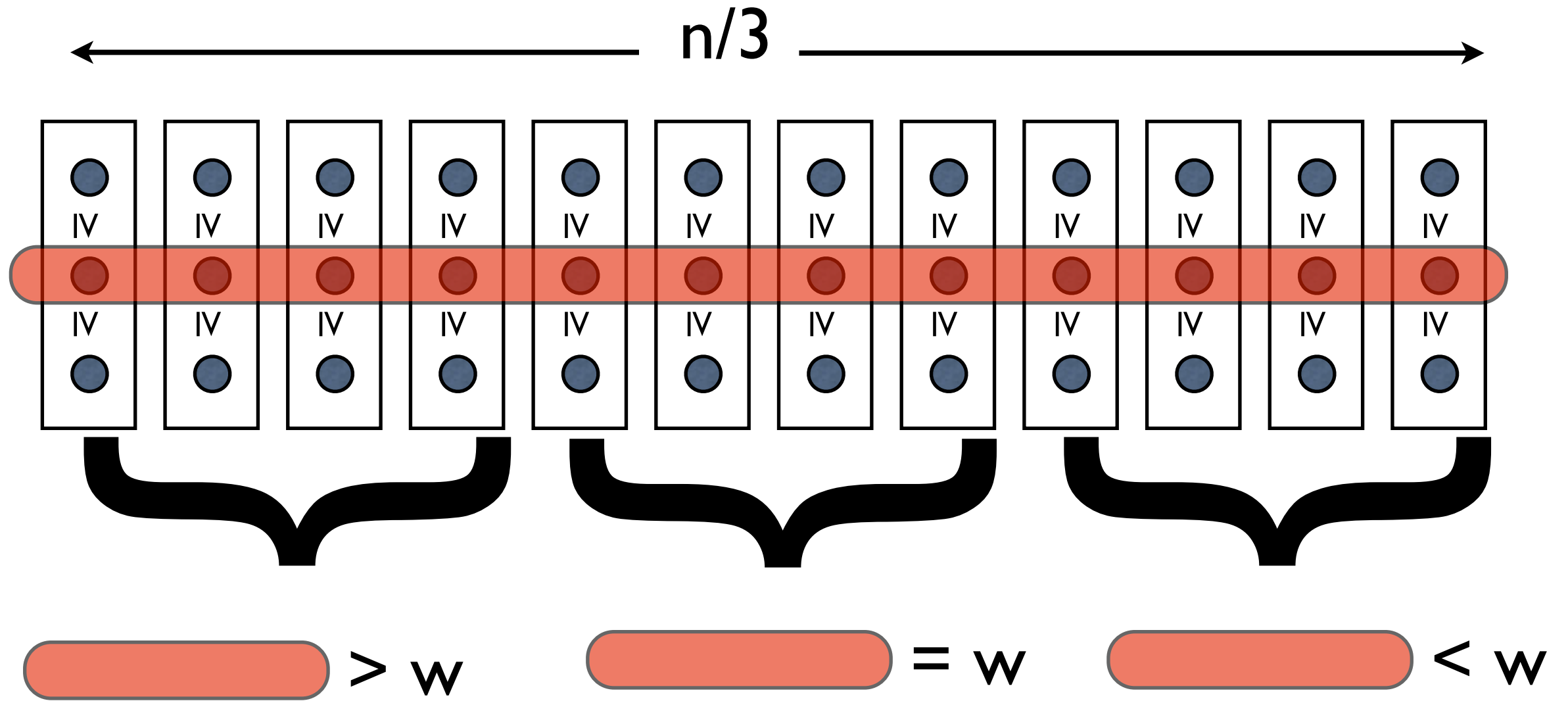
$$S_E(w) = \{x_i \mid x_i = w\}$$

$$S_G(w) = \{x_i \mid x_i > w\}$$

} Can be computed in linear time

4. if $k \leq |S_L(w)|$, output $\text{Sel}(S_L(w), k)$
 else if $k \leq |S_L(w)| + |S_E(w)|$, output w
 else output $\text{Sel}(S_G(w), k - |S_L(w)| - |S_E(w)|)$

Algorithm: Sel(numbers,k)



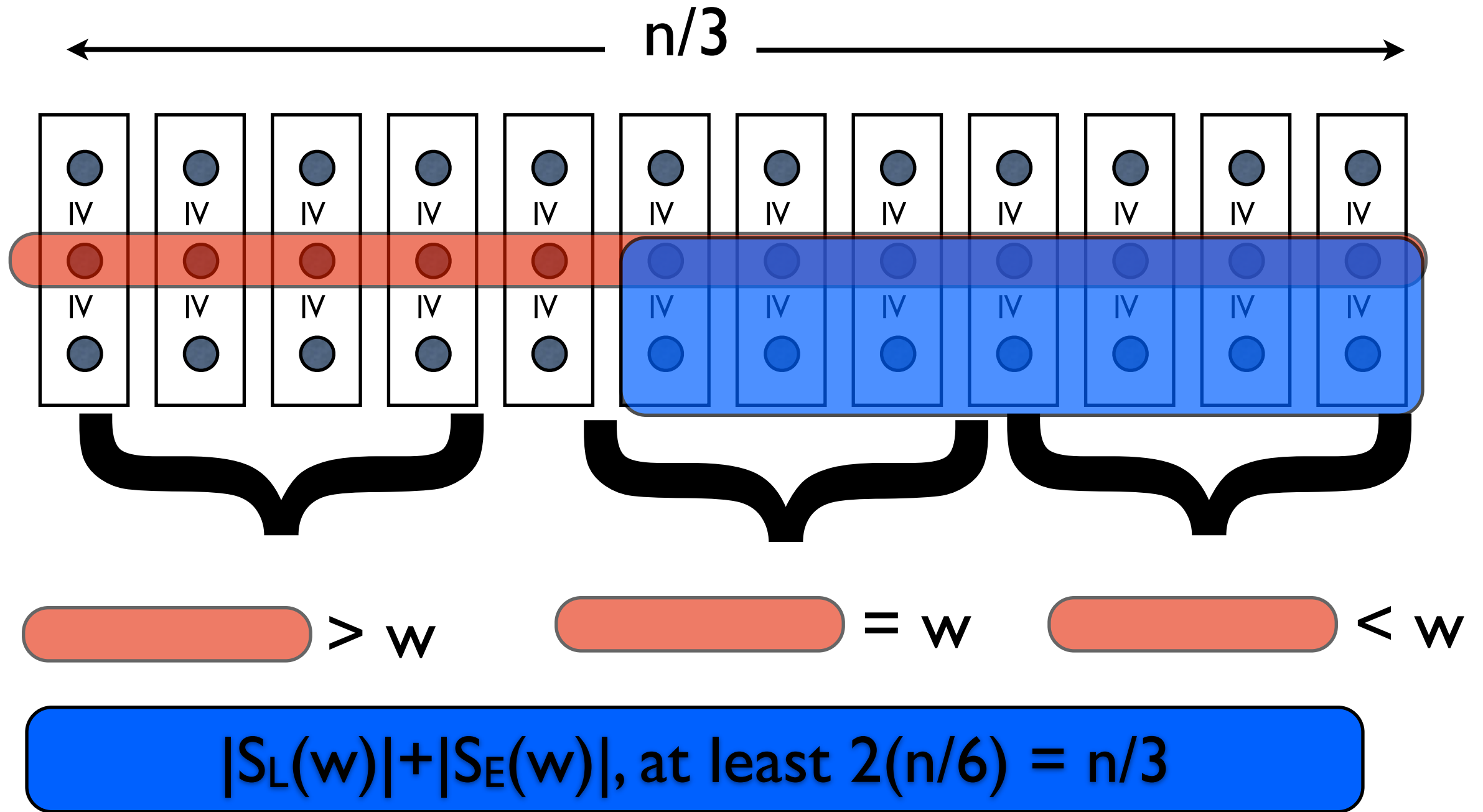
$$w = \text{Sel}(\text{red bar}, n/6)$$

if $k \leq |S_L(w)|$, output $\text{Sel}(S_L(w), k)$

else if $k \leq |S_L(w)| + |S_E(w)|$, output w

else output $\text{Sel}(S_G(w), k - |S_L(w)| - |S_G(w)|)$

Algorithm: Sel(numbers,k)



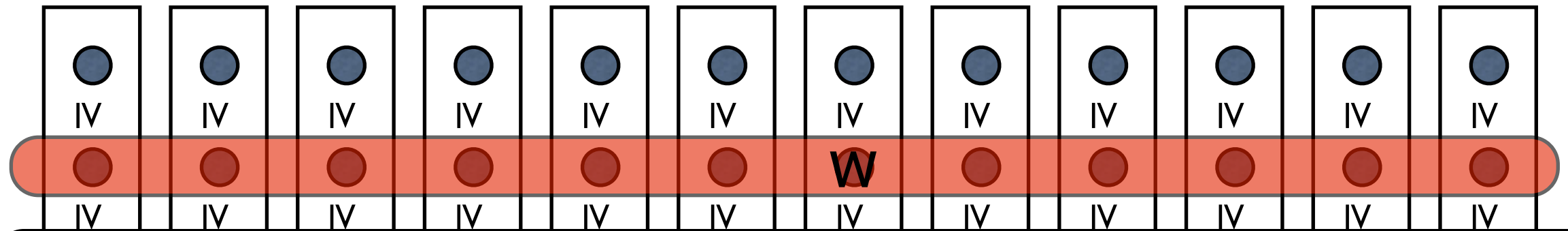
$$w = \text{Sel}(\text{red bar}, n/6)$$

if $k \leq |S_L(w)|$, output Sel($S_L(w)$, k)

else if $k \leq |S_L(w)| + |S_E(w)|$, output w

else output Sel($S_G(w)$, $k - |S_L(w)| - |S_G(w)|$)

← $n/3$ →



$|S_L(w)| + |S_E(w)|, |S_G(w)| + |S_E(w)|, \text{ at least } 2(n/6) = n/3$

1. Partition numbers into sets of size 3
2. Sort each set
3. $w = \text{median of } \dots$

$T(n) = T(n/3) + T(2n/3) + O(n)$

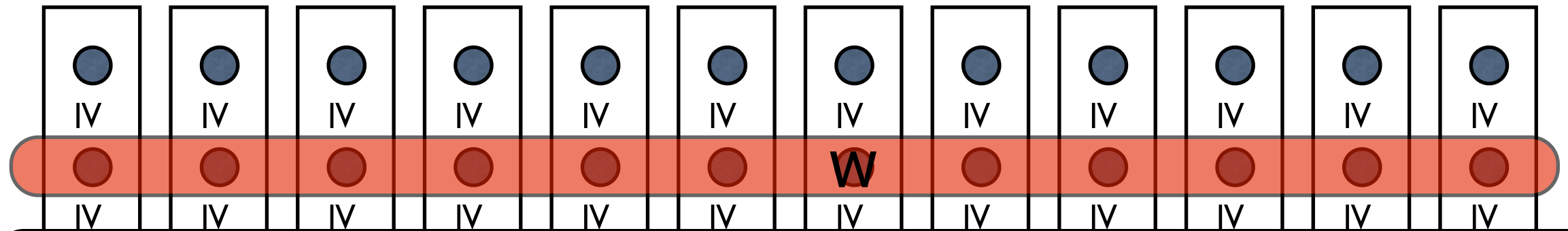
$$\begin{aligned}
 S_L(w) &= \{x_i \mid x_i < w\} \\
 S_E(w) &= \{x_i \mid x_i = w\} \\
 S_G(w) &= \{x_i \mid x_i > w\}
 \end{aligned}
 \left. \vphantom{\begin{aligned} S_L(w) \\ S_E(w) \\ S_G(w) \end{aligned}} \right\}$$

Can be computed in linear time

4. if $k \leq |S_L(w)|$, output $\text{Sel}(S_L(w), k)$
 else if $k \leq |S_L(w)| + |S_E(w)|$, output w
 else output $\text{Sel}(S_G(w), k - |S_L(w)| - |S_E(w)|)$

Algorithm: Sel(numbers, k)

← $n/3$ →



$|S_L(w)| + |S_E(w)|, |S_G(w)| + |S_E(w)|$, at least $2(n/6) = n/3$

Algorithm: Sel(numbers, k)

1. Partition numbers into sets of size 3
2. Sort each set
3. $w = \text{median}$

$T(n) = T(n/3) + T(2n/3) + O(n)$
 so
 $T(n) = O(n \log n)$
 (what's the point???)

$$S_L(w) = \{x_i \mid x_i < w\}$$

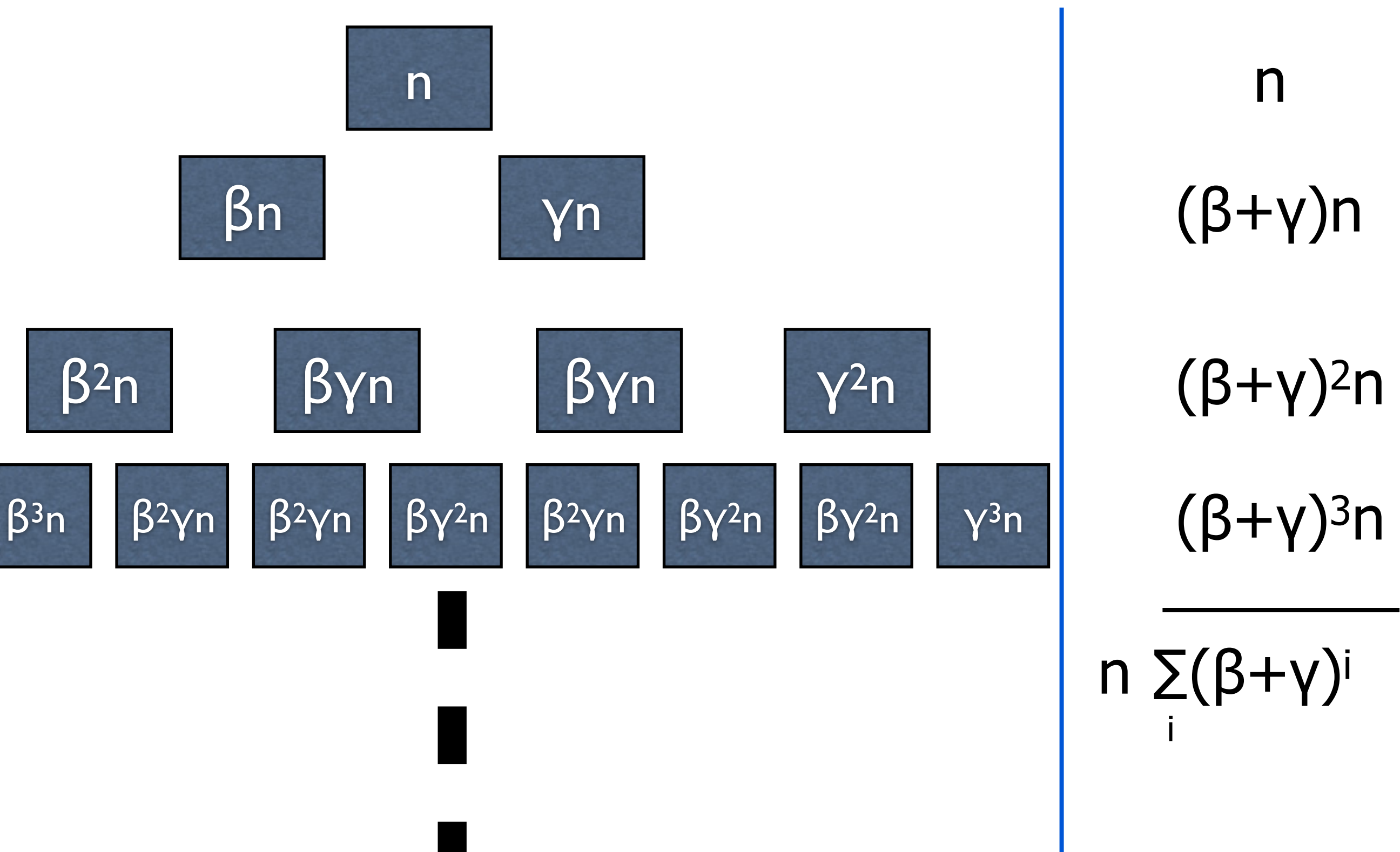
$$S_E(w) = \{x_i \mid x_i = w\}$$

$$S_G(w) = \{x_i \mid x_i > w\}$$

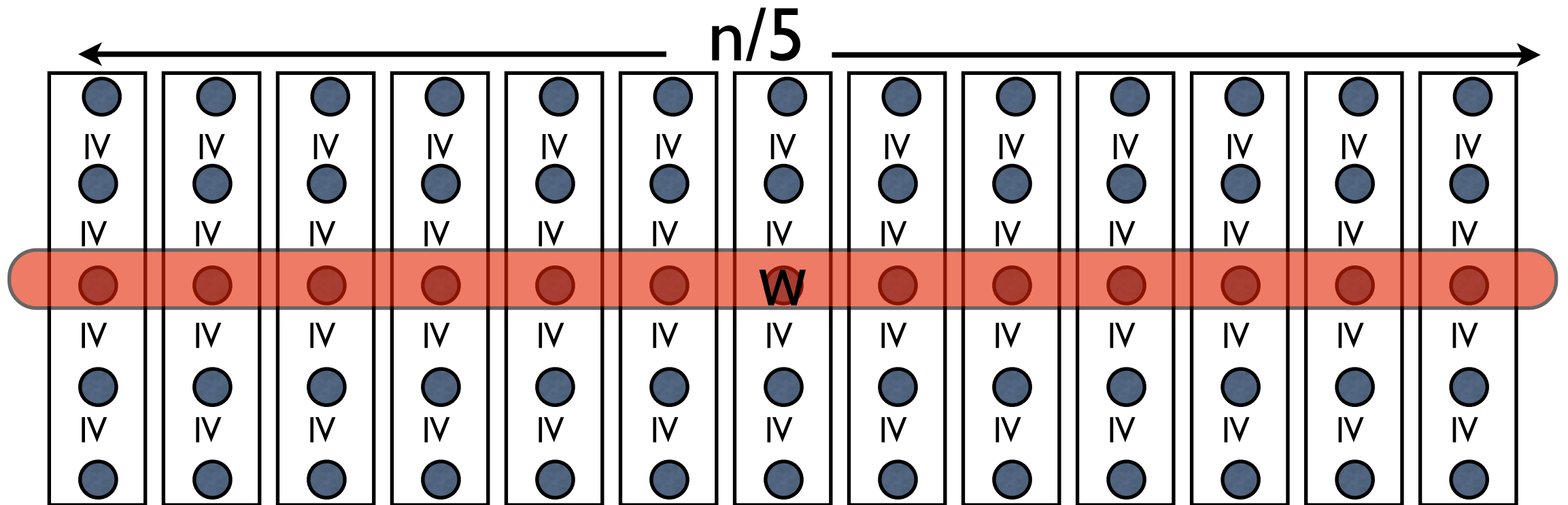
4. if $k \leq |S_L(w)|$, output Sel($S_L(w)$, k)
 else if $k \leq |S_L(w)| + |S_E(w)|$, output w
 else output Sel($S_G(w)$, $k - |S_L(w)| - |S_E(w)|$)


Recurrences

$$T(n) = T(\gamma n) + T(\beta n) + n$$



Algorithm: Sel(numbers,k)



1. Partition numbers into sets of size 5
2. Sort each set
3. $w = \text{median of}$ 

$$S_L(w) = \{x_i \mid x_i < w\}$$

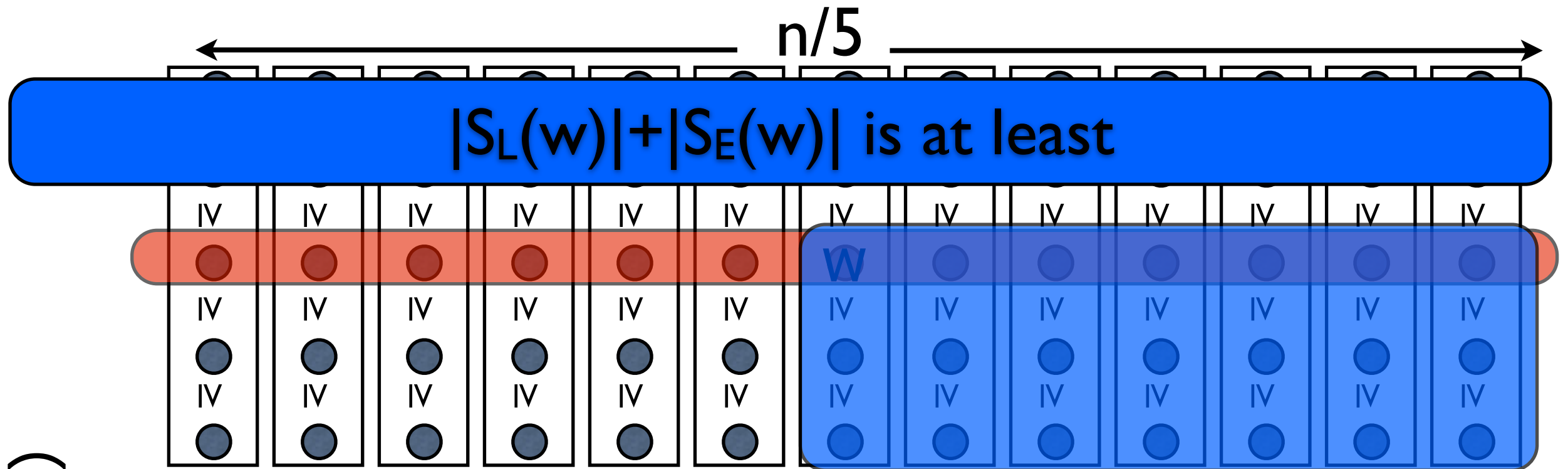
$$S_E(w) = \{x_i \mid x_i = w\}$$


$$S_G(w) = \{x_i \mid x_i > w\}$$

} Can be computed in linear time

4. if $k \leq |S_L(w)|$, output $\text{Sel}(S_L(w), k)$
else if $k \leq |S_L(w)| + |S_E(w)|$, output w
else output $\text{Sel}(S_G(w), k - |S_L(w)| - |S_E(w)|)$

Algorithm: Sel(numbers,k)



1. Partition numbers into sets of size 5
2. Sort each set
3. $w = \text{median of}$ 

$$S_L(w) = \{x_i \mid x_i < w\}$$

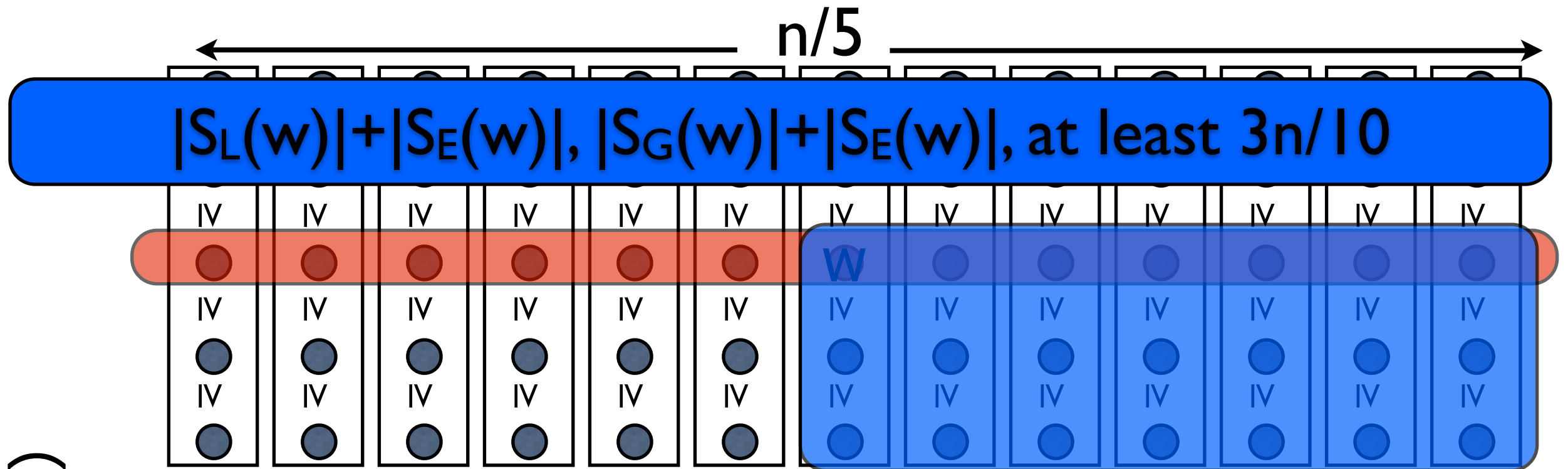
$$S_E(w) = \{x_i \mid x_i = w\}$$


$$S_G(w) = \{x_i \mid x_i > w\}$$

} Can be computed in linear time

4. if $k \leq |S_L(w)|$, output $\text{Sel}(S_L(w), k)$
else if $k \leq |S_L(w)| + |S_E(w)|$, output w
else output $\text{Sel}(S_G(w), k - |S_L(w)| - |S_E(w)|)$

Algorithm: Sel(numbers,k)



1. Partition numbers into sets of size 5
2. Sort each set
3. $w = \text{median of}$ 

$$S_L(w) = \{x_i \mid x_i < w\}$$

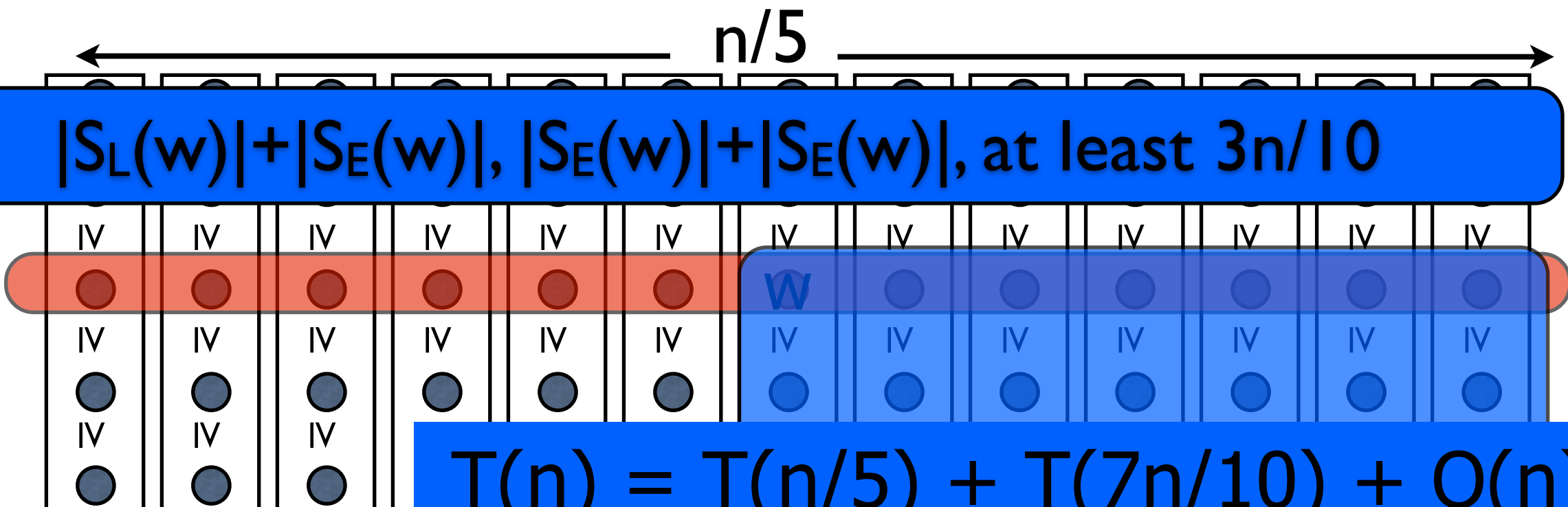
$$S_E(w) = \{x_i \mid x_i = w\}$$

$$S_G(w) = \{x_i \mid x_i > w\}$$

} Can be computed in linear time

4. if $k \leq |S_L(w)|$, output $\text{Sel}(S_L(w), k)$
else if $k \leq |S_L(w)| + |S_E(w)|$, output w
else output $\text{Sel}(S_G(w), k - |S_L(w)| - |S_E(w)|)$

Algorithm: Sel(numbers,k)



1. Partition numbers into sub-arrays of size $n/5$
2. Sort each sub-array
3. $w = \text{median}$ of the middle sub-array

$T(n) = T(n/5) + T(7n/10) + O(n)$

so

$T(n) = O(n)$

$S_L(w) = \{x_i \mid x_i < w\}$

$S_E(w) = \{x_i \mid x_i = w\}$

$S_G(w) = \{x_i \mid x_i > w\}$

} Can be computed in linear time

4. if $k \leq |S_L(w)|$, output Sel($S_L(w)$, k)
else if $k \leq |S_L(w)| + |S_E(w)|$, output w
else output Sel($S_G(w)$, $k - |S_L(w)| - |S_E(w)|$)