

Sorting

* Bubble Sort $O(n^2)$
 * Insertion Sort $O(n^2)$
 * Quick Sort $\rightarrow O(n \log n)$
 * Merge Sort $\rightarrow O(n \log n)$

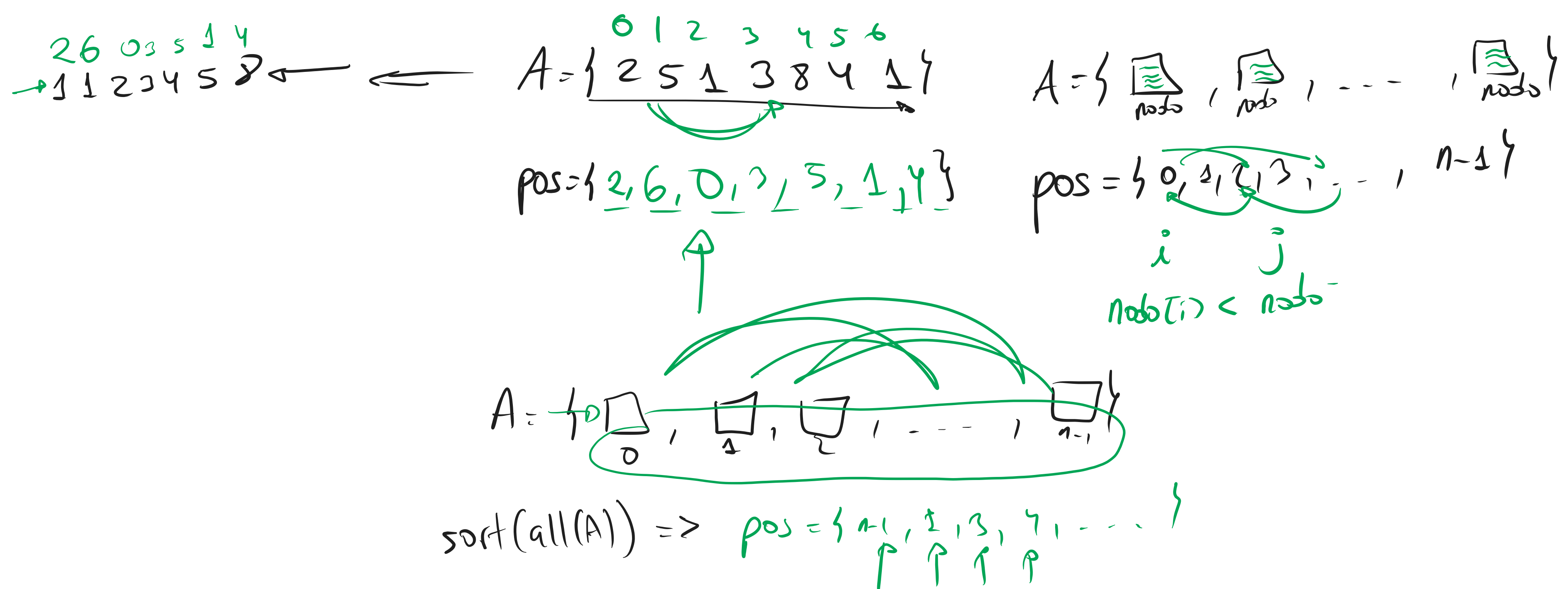
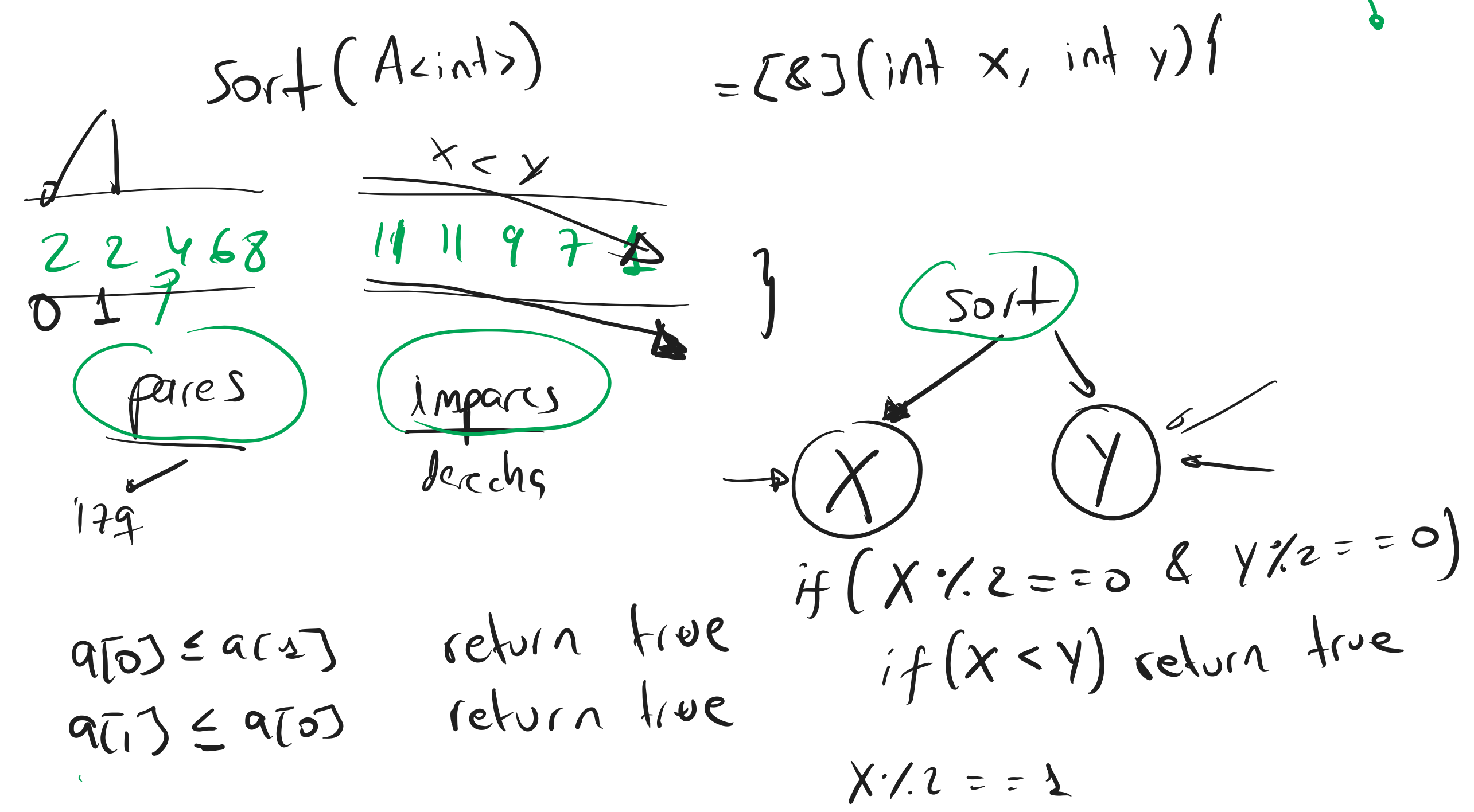
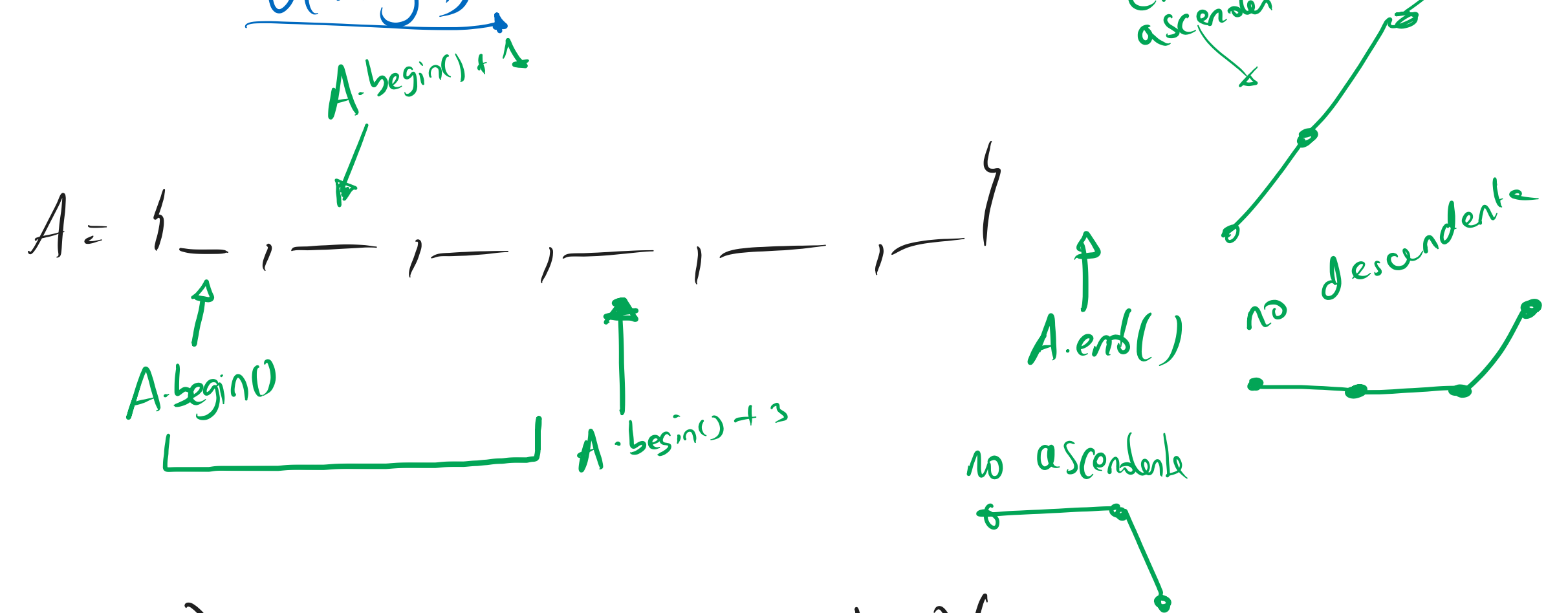
$A = \{2, 3, 1, 5, 8, 3, 6, 4\}$

$(4, 2, 3, 5) \quad (3, 4, 6, 8)$

$A = \{1, 1, 2, 3, 3, 4, 5, 6, 8\}$

$$T(n) = 2T(n/2) + O(n) \rightarrow T(n) = O(n \log n)$$

* Propiedad: Usando un algoritmo de comparación, la complejidad mínima de un ordenamiento es $O(n \log n)$



Dado un arreglo A de n elementos

Consulta: Número k \rightarrow La posición del k-th elemento más pequeño

$A = \{3, 1, 5, 6, 2, 9, 8, 3\}$

$k=4 \Rightarrow 7$
 $k=5 \Rightarrow 2$

$A = \{2, 2, 3, 3, 4, 4, 7, 2, 1\}$

Algoritmos de Ord. sin comparar

* Counting Sort $O(n)$

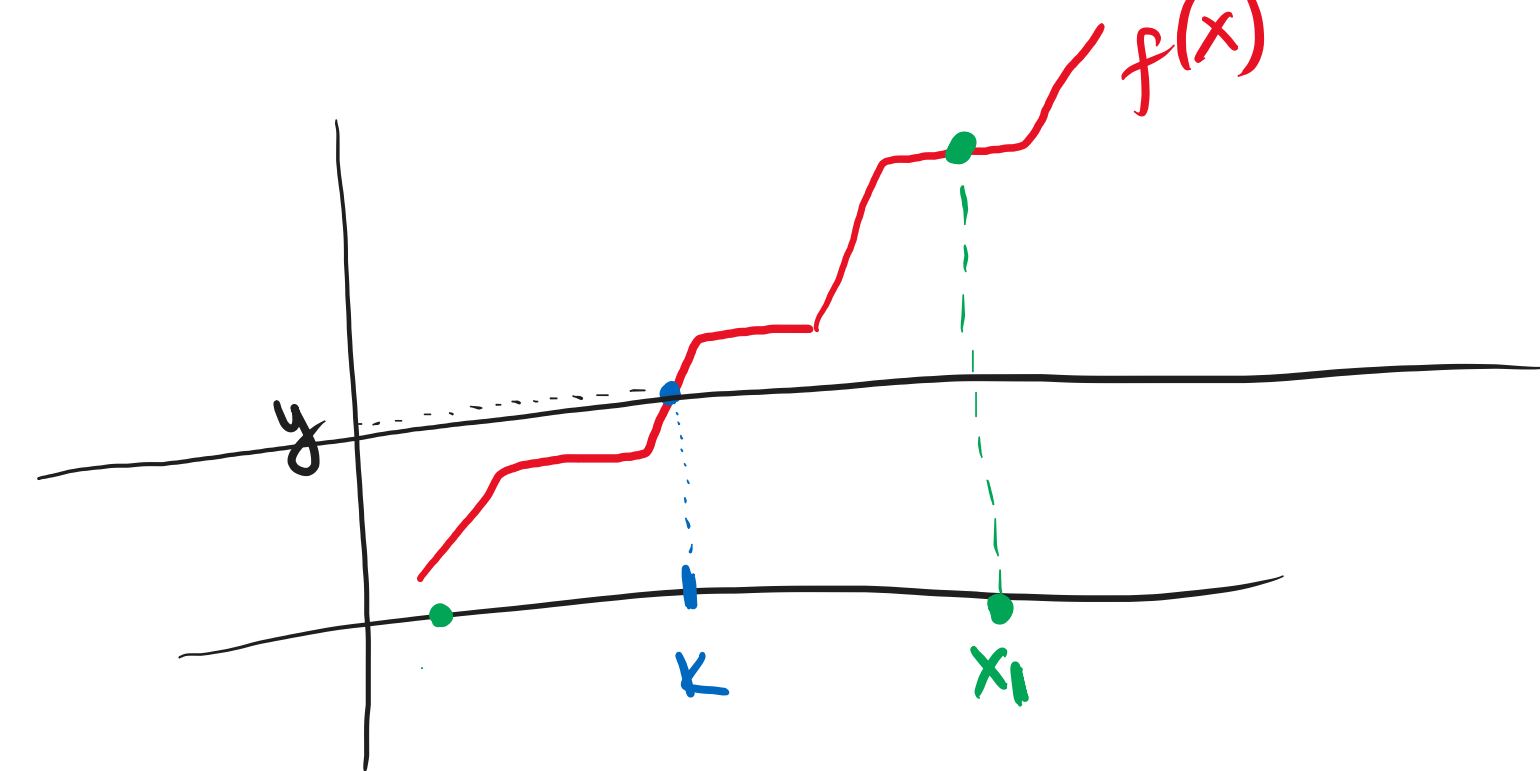
$s = abcabdcbaehca$
 $s = aaabbbcccddeh$

$a \rightarrow 0$
 $b \rightarrow 1$
 $c \rightarrow 25$

Búsqueda Binaria

* Funciones Monótonas

min k / $f(x) = y$

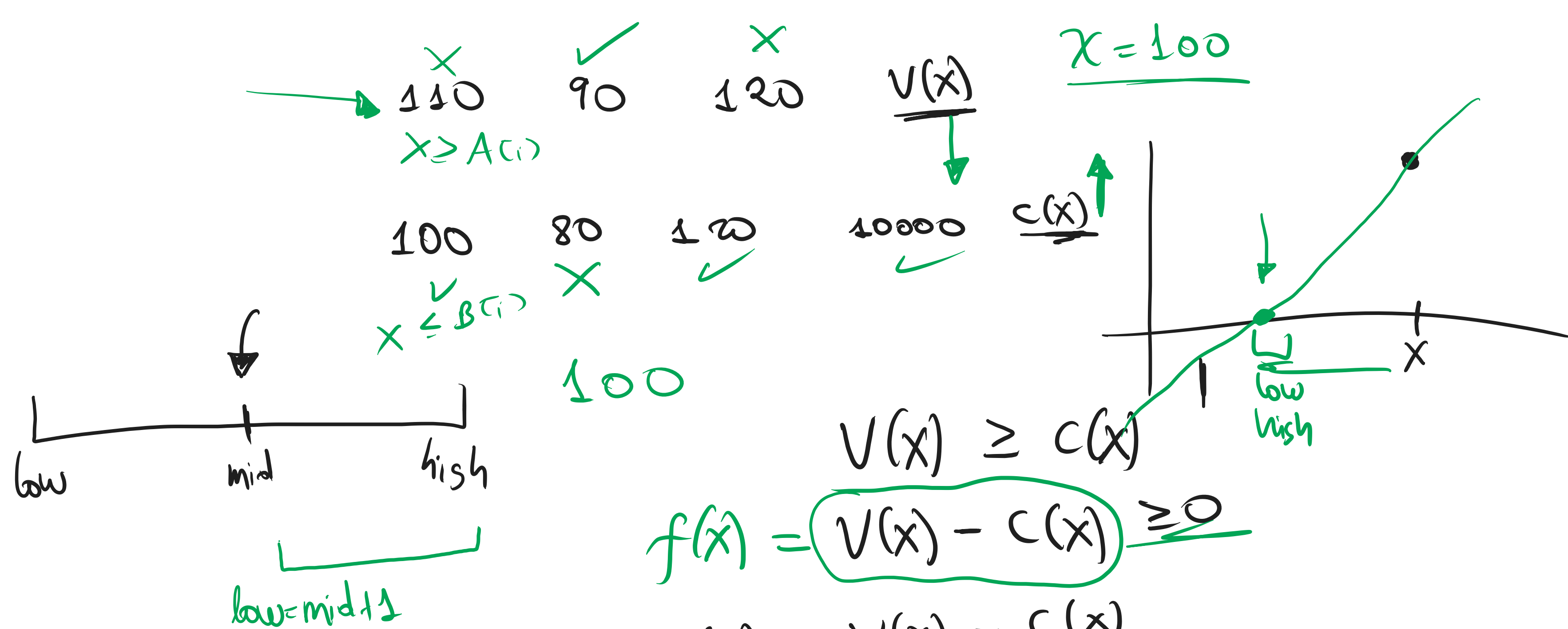


low mid high

Variables: $1, 2, \dots, N$

Comparadores: $\leq B_1, \leq B_2, \leq B_3, \dots, \leq B_m$

$(8+2)/2 = 5$



$$f(x) = V(x) - C(x) \geq 0$$

$$f(x) = V(x) - C(x)$$

$$f(x_1) = V(x_1) - C(x_1)$$

$$V(x_1) \leq V(x) \quad -C(x_1) \leq -C(x) \quad \Rightarrow \quad V(x_1) - C(x_1) \leq V(x) - C(x)$$

* Función búsqueda: monótona

* H

$H \rightarrow \frac{H}{2} \rightarrow \frac{H}{4} \rightarrow \frac{H}{8} \dots \rightarrow 1$

$\Rightarrow O(\log H) \quad O(\log_{10} N)$

low mid high