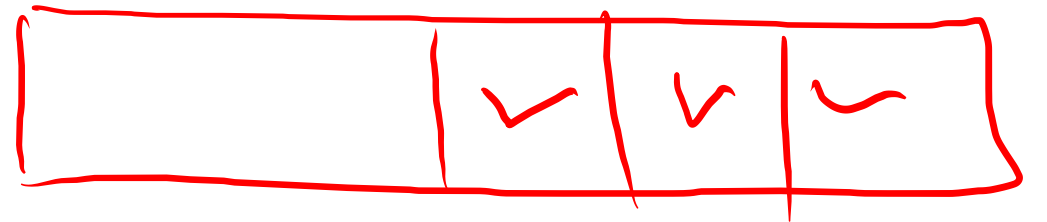
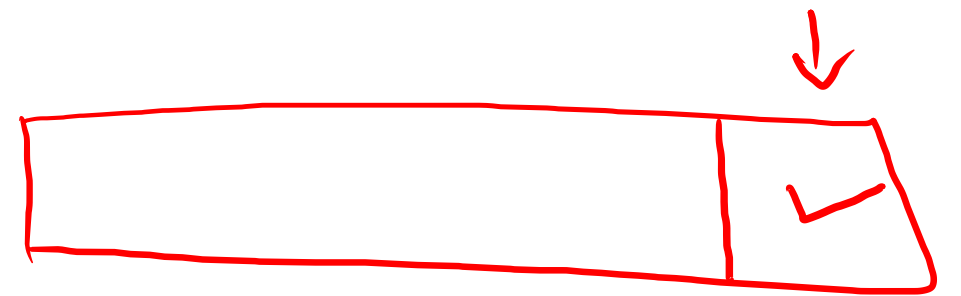
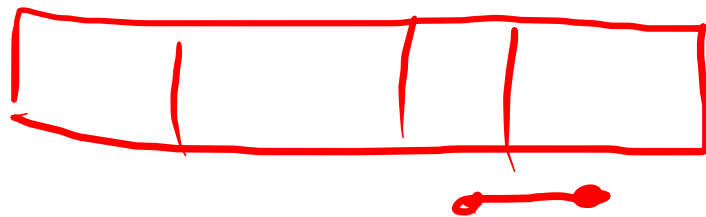
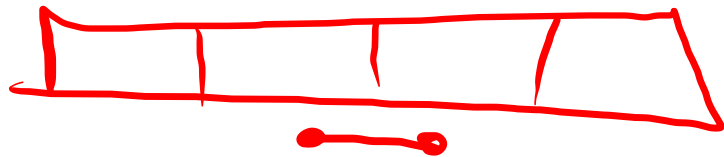
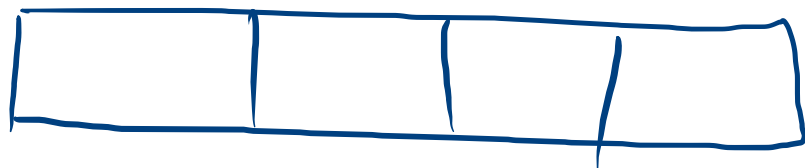


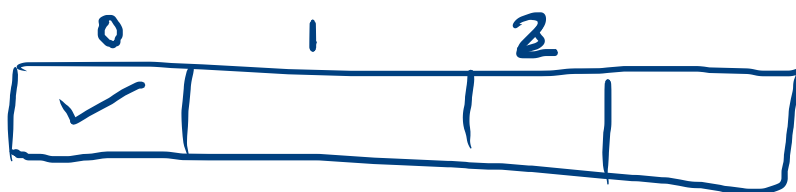
Selection



n



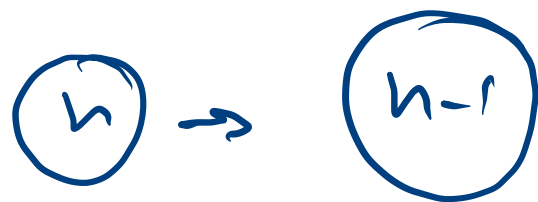
1 iter



2 iter



3 iter



i ←

select smaller

0 7 -2 4 1 3

min1

1✓

-2

7

4

1 3
min1

2✓

-2

1

4

7 3

```

int n = arr.length;
for(int i=0; i<n-1; i++){
    int minI = i;
    for(int j=i+1; j<n; j++){
        if(isSmaller(arr, j, minI)){
            minI = j;
        }
    }
    swap(arr, i, minI);
}

```

$i=0$

0
7

1
-2

minI

2
4

3
1

4
3

$i=1$

-2

7

4

1 3

minI

$i=2$

-2 1

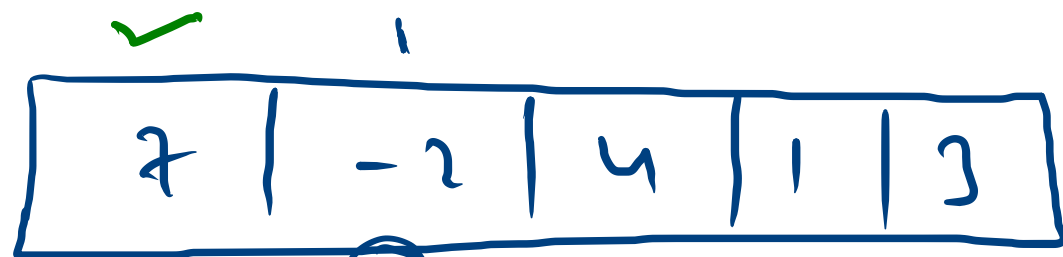
4

7 3

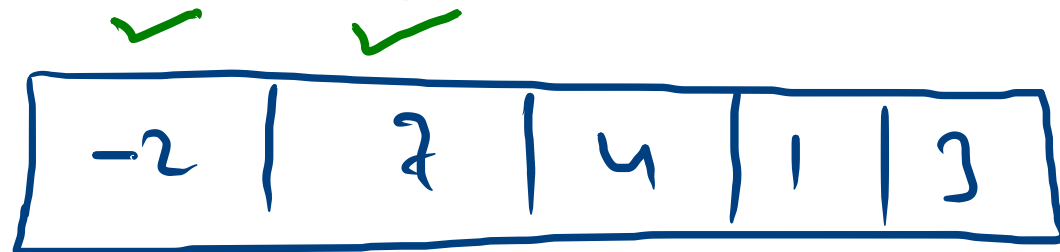
minI

$i=3$

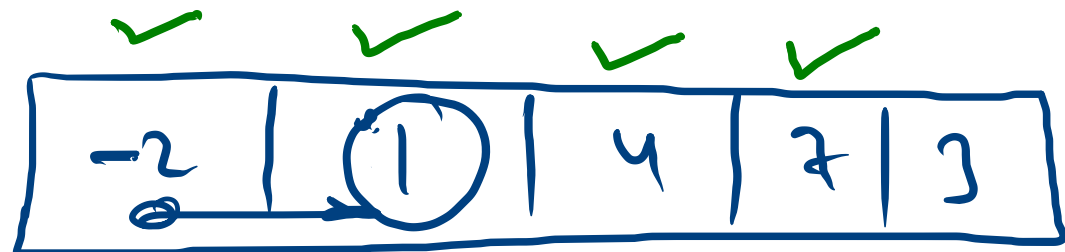
minI



i

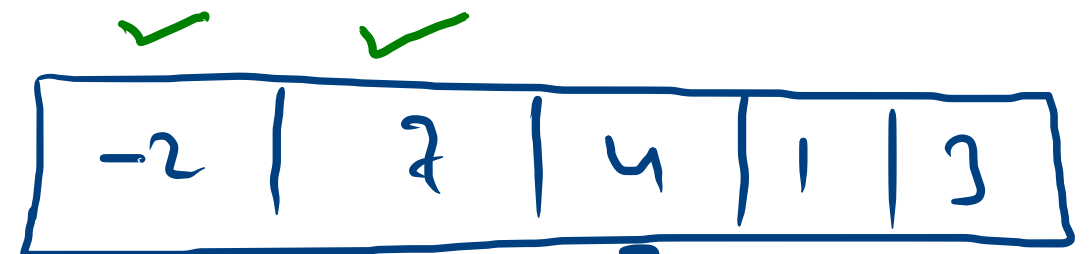


i



j

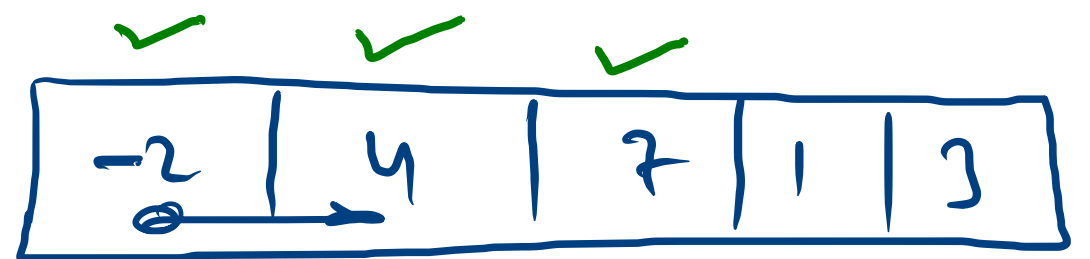
i



j

i

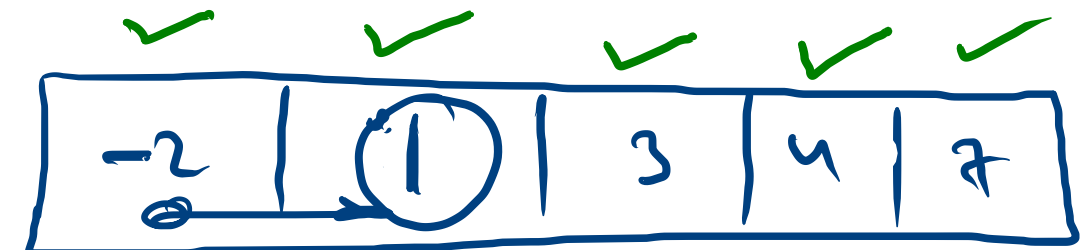
j+1



j

j+1

i

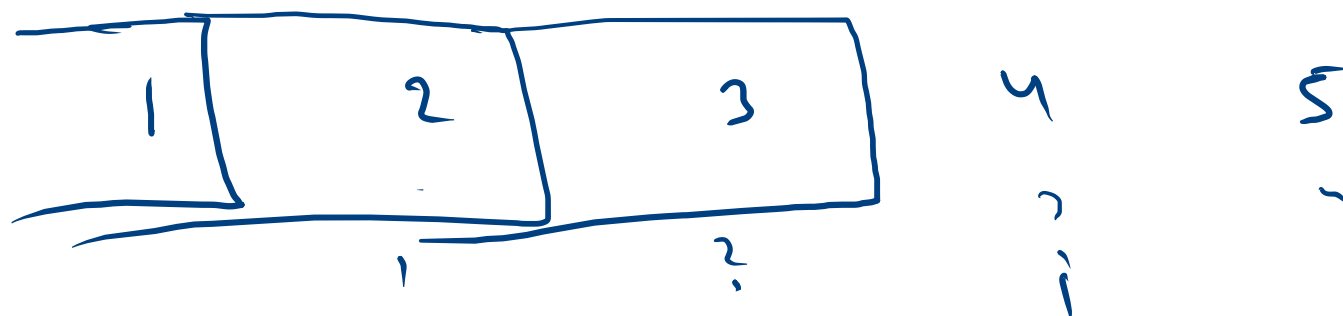


j

i

break

$\Omega(n)$ \leftarrow ~~sort~~ sort

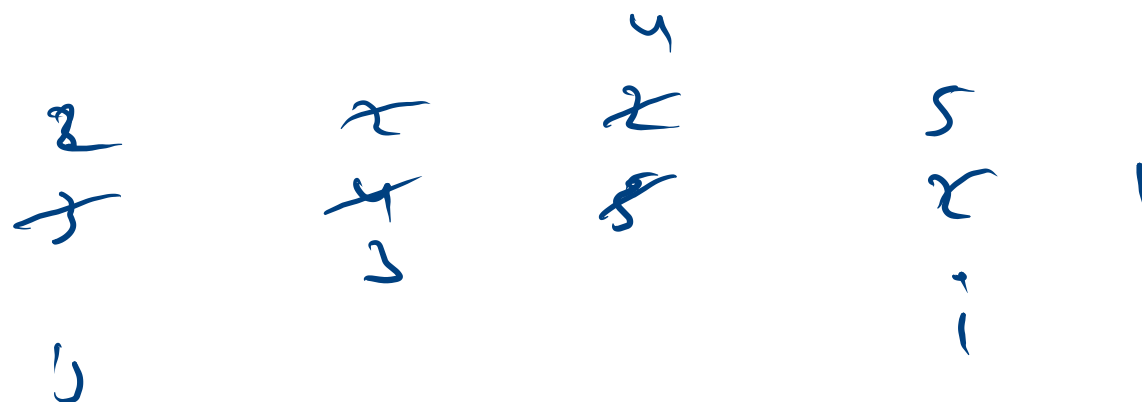


(K)

```
for(int i=1; i<arr.length; i++){
    for(int j=i-1; j>=0; j--){
        if(isGreater(arr, j, j+1)){
            swap(arr, j, j+1);
        } else {
            break;
        }
    }
}
```

i = 1
2
3
4
5
n-1

①
②
③
4
5
n-1



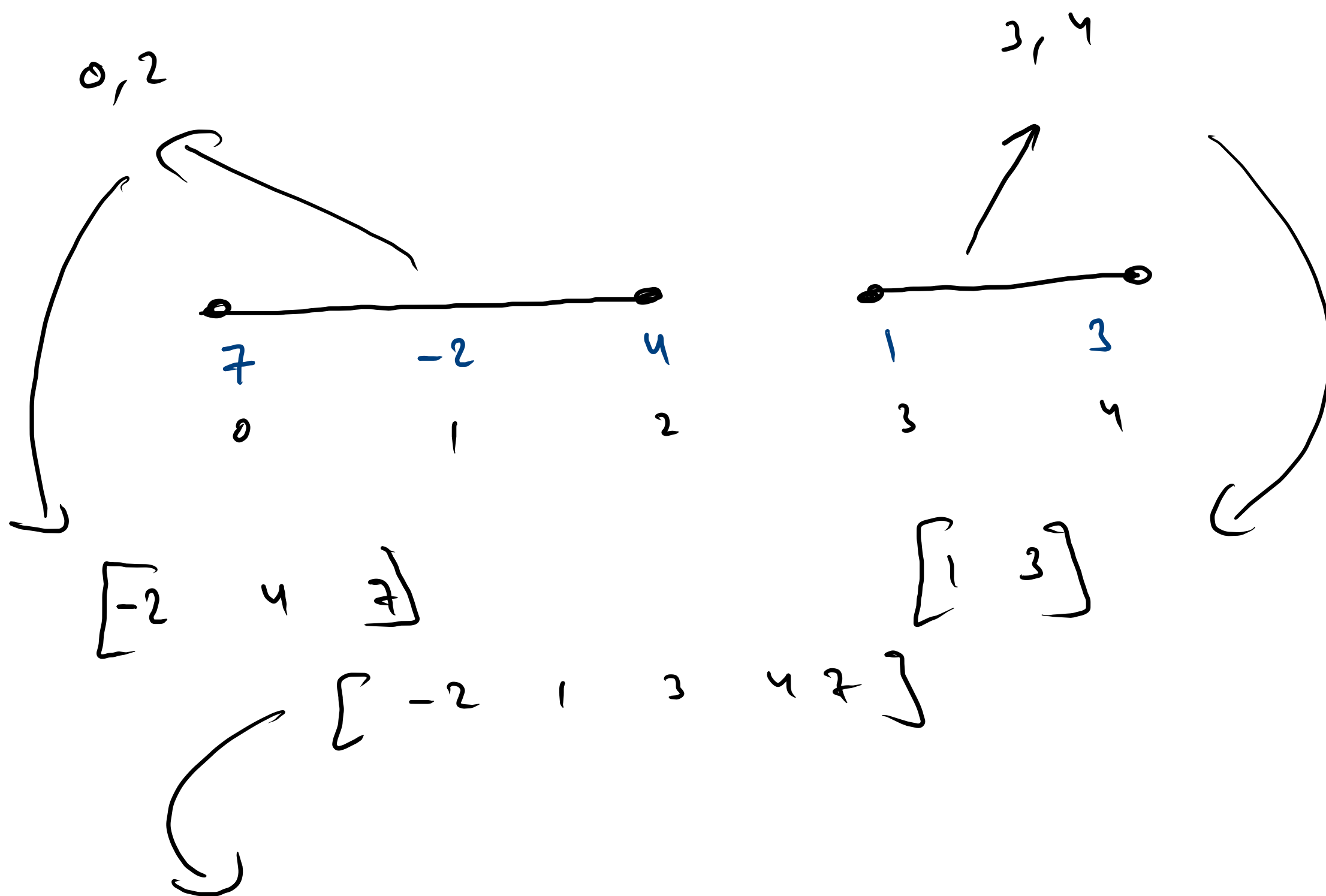
$$1 + 2 + 3 + \dots + n-1$$

$$\frac{(n-1) \cdot n}{2} \rightarrow \frac{n^2 - n}{2} = \boxed{\Theta(n^2)}$$

$a \rightarrow [\overset{\bullet}{\underset{\bullet}{4}} \overset{\bullet}{7} \overset{\bullet}{\boxed{8 \ 9 \ 12}}]$
 $b \rightarrow [\overset{\bullet}{1} \overset{\bullet}{2} \overset{\bullet}{3} \overset{\bullet}{6} \overset{\bullet}{7} \overset{\bullet}{\boxed{7}}]$

$[1 \ 2 \ 4 \ 5 \ 6 \ 7 \ 7 \ 8 \ 12]$

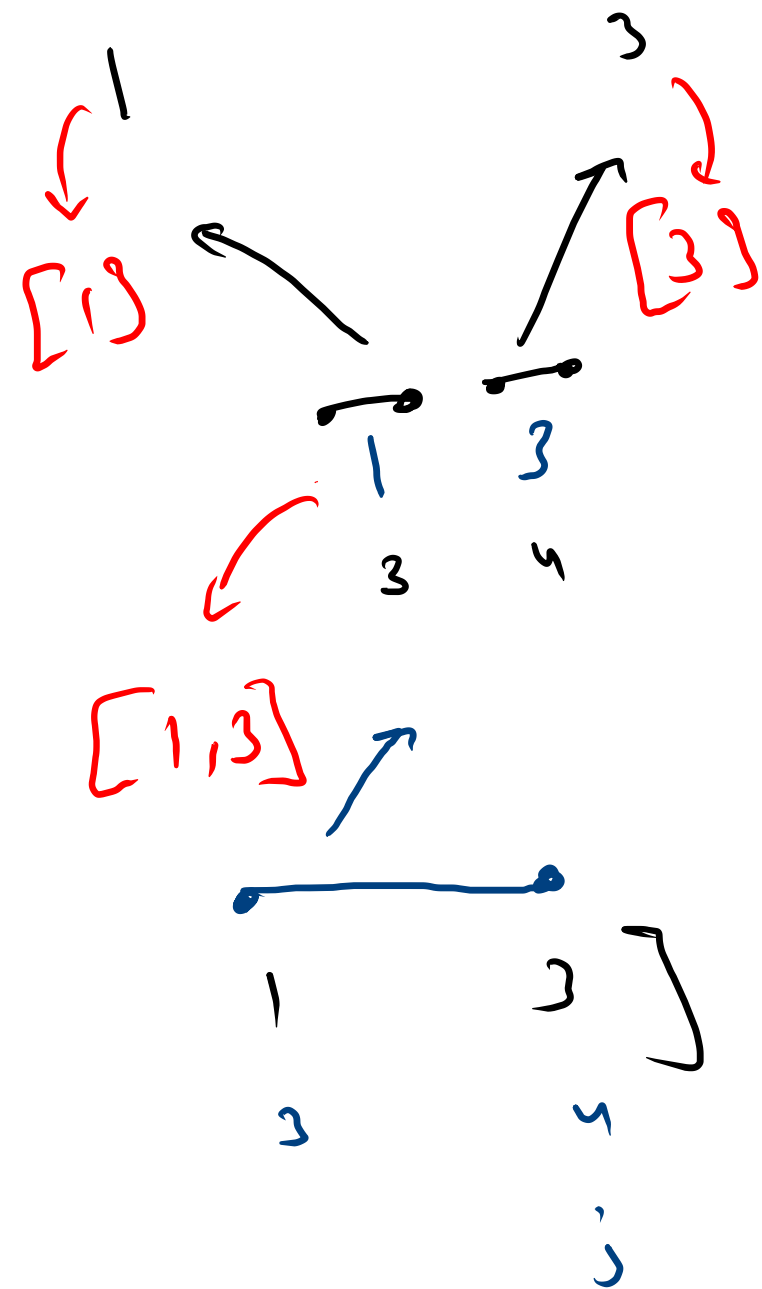
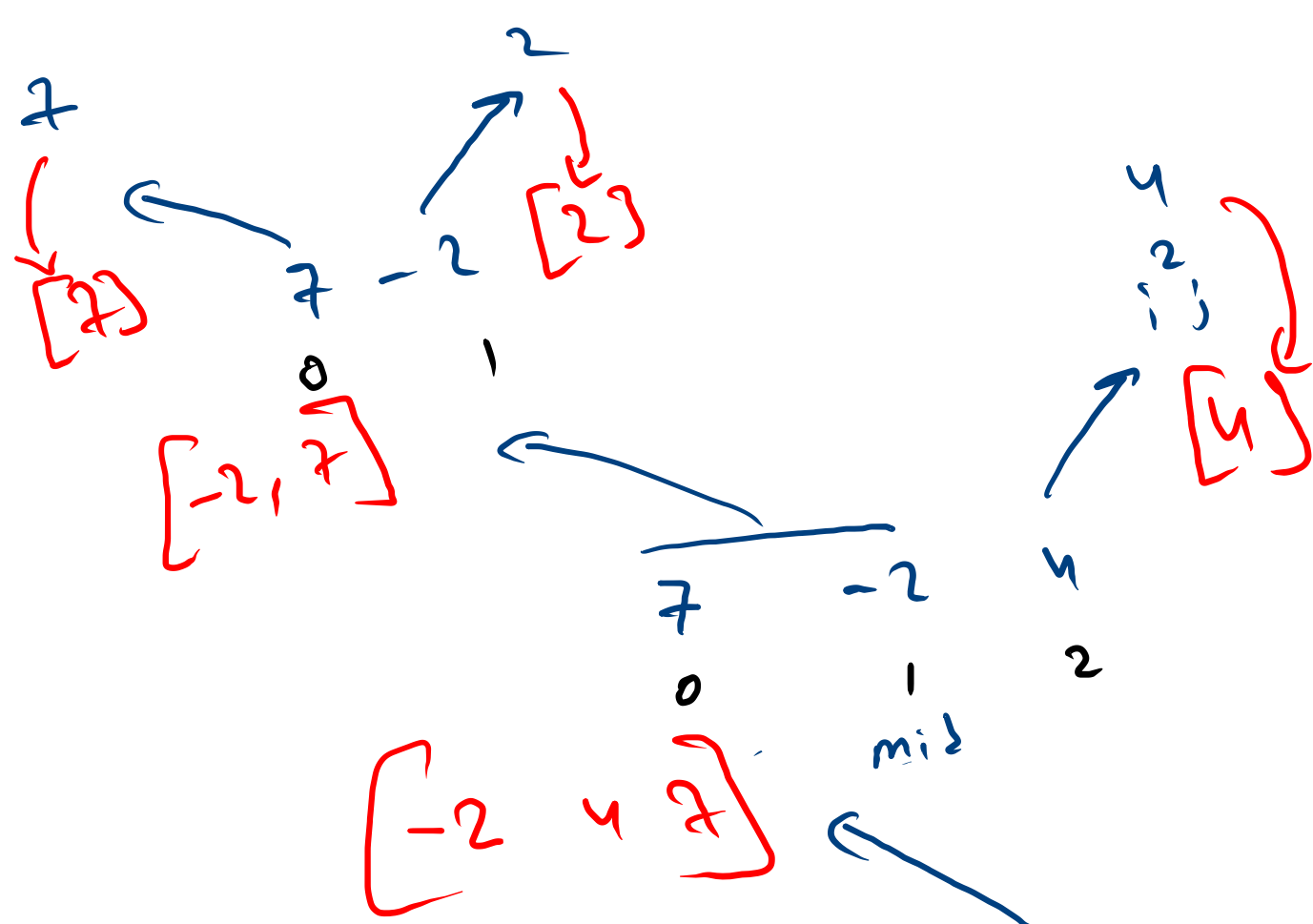
(a, b) $[1 \ 2 \ 4 \ 5 \ 6 \ \boxed{7 \ 7} \ 8 \ 9 \ 12]$



7 -2 4 1 3

arr →

hash → [-2 1 3 4 7]



3+4
7
3


```

public static int[] mergeSort(int[] arr, int lo, int hi) {
    if (lo == hi) {
        int ans[] = new int[1];
        ans[0] = arr[lo];
        return ans;
    }

    int mid = (lo+hi)/2;
    int left[] = mergeSort(arr, lo, mid);
    int right[] = mergeSort(arr, mid+1, hi);

    int ans[] = mergeTwoSortedArrays(left, right);

    return ans;
}

```

$$m = \log_2(n)$$

$$f(n) = (m+1)n$$

$$= mn + n$$

$$g(n) = mn + mn = 2mn$$

$$O(mn) = O(n \log_2(n))$$

$$\text{exact time} = (n \log(n) + n)$$

iterative
 recursion
 ↓
 higher level build

work

$$+ \textcircled{n}$$

$$+ \textcircled{n} \times 2$$

$$+ \textcircled{n} \times 4$$

$$T(n) = \textcircled{2} T\left(\frac{n}{2}\right)$$

$$\cancel{2 T\left(\frac{n}{2}\right)} = \cancel{4 T\left(\frac{n}{4}\right)}$$

$$\cancel{4 T\left(\frac{n}{4}\right)} = \cancel{8 T\left(\frac{n}{8}\right)}$$

⋮

$$\cancel{T(1)}$$

$$1$$

$$O(n \log(n))$$

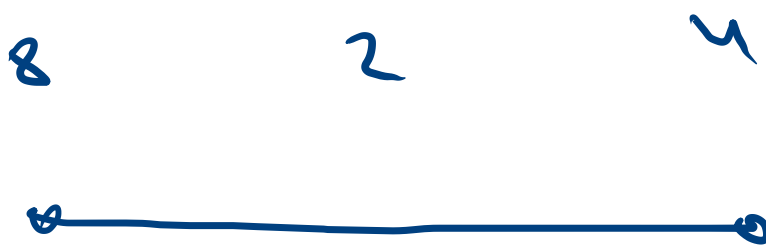
$$\begin{aligned}
 m &= \log_2(n) \\
 (m+1) \times n \\
 (\log_2(n)+1) \times n \\
 \theta(n \log_2(n))
 \end{aligned}$$

$$\begin{aligned}
 2^0 \quad T(n) &= 2T\left(\frac{n}{2}\right) + \textcircled{n} \\
 2^1 \quad 2T\left(\frac{n}{2}\right) &= 4T\left(\frac{n}{4}\right) + \frac{\textcircled{n} \times 2}{2} \\
 2^2 \quad 4T\left(\frac{n}{4}\right) &= 8T\left(\frac{n}{8}\right) + \frac{\textcircled{n} \times 4}{4} \\
 &\vdots \\
 2^m \quad T(1) &= 1 \times 2^m = n
 \end{aligned}$$

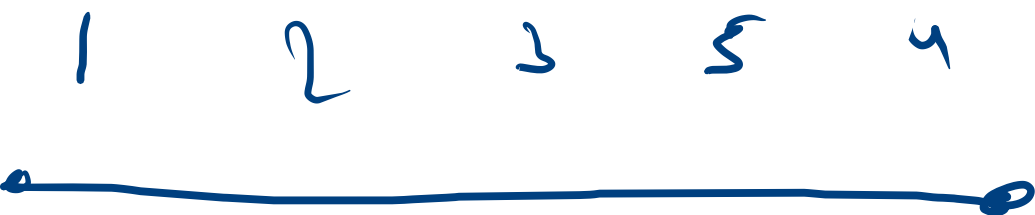
Pivot = 6



\leq pivot
6



$>$ pivot
6



\leq pivot



$>$ pivot

pivot 5

							j-1	i		
0										
1	2	4	5	3	1	2	8	8	7	2
							<u>8 8 7 2 9</u>			
							big			
small, equal										

0 ... j-1 - \leq pivot
j ... i-1 - $>$ pivot
i \rightarrow unknown

```
if (arr[i] > pivot)
    i++
else
    swap(j, i)
    j++
    i++
```

i	= 0
j	= 0

```
int i=0;
int j=0;
```

```
while(i<arr.length){
    if(arr[i] <= pivot){
        swap(arr, i, j);
        j++;
    }
    i++;
}
```

0 ... j-1 - \leq pivot
 j ... i-1 - $>$ pivot
 i → unknown

pivot → 5 i

5 2 6
~~5~~ ~~2~~

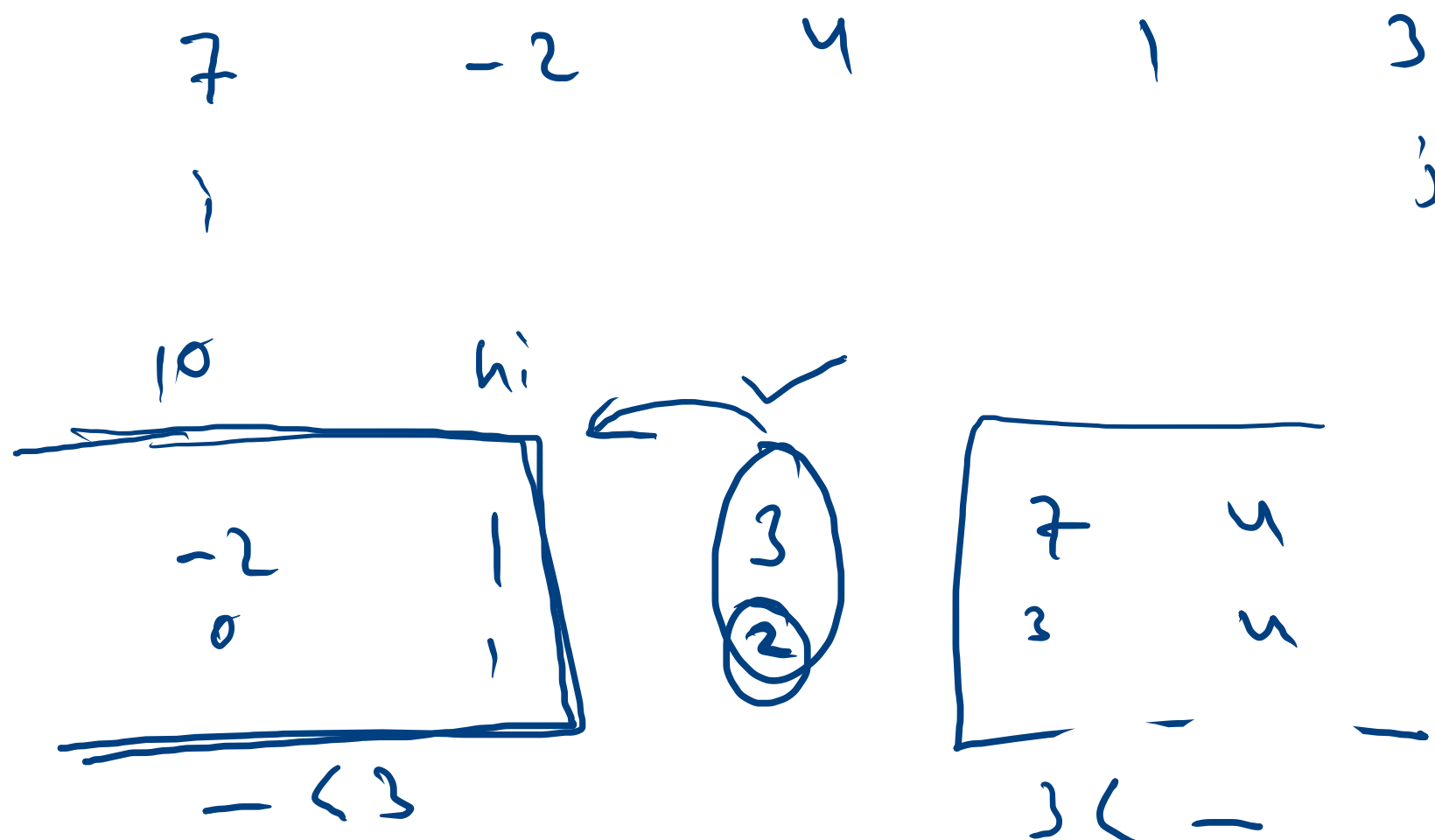
5 2 6 3
~~5~~

$O(n)$

h.k → $O(n)$

7 -2 4 1 3
 $k \Rightarrow 3$

$k = 2$
 $k = 1$



int pivotI = partition(arr, lo, hi, ³pivot)

```
public static int quickSelect(int[] arr, int lo, int hi, int k) {
    int pivot = arr[hi];

    int pivotInd = partition(arr, pivot, lo, hi); 1

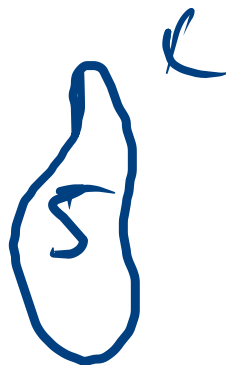
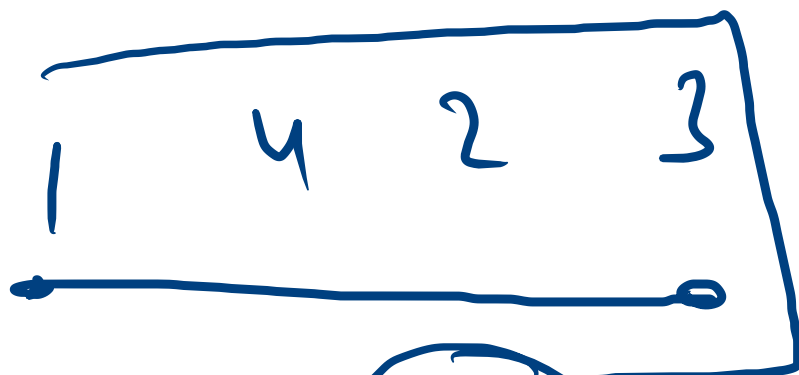
    if(pivotInd == k){
        return arr[pivotInd];
    } 4 < 2
    if(k < pivotInd){
        return quickSelect(arr, lo, pivotInd-1, k);
    }else{// k > pivotInd
        return quickSelect(arr, pivotInd+1, hi, k);
    }
}
```

$\begin{matrix} 7 & -2 & 4 & 1 & \textcircled{3} \\ 10 & & & & h_i \end{matrix}$

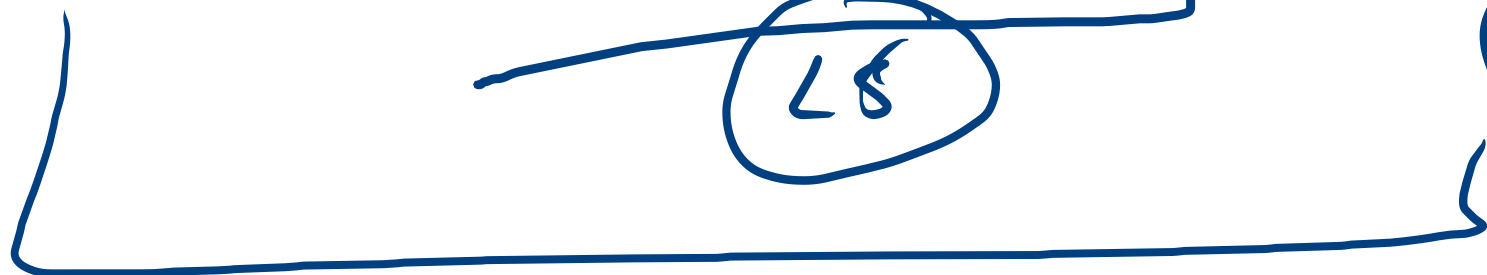
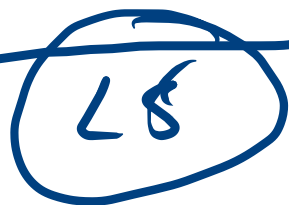
A handwritten diagram of a 2D coordinate system. The horizontal axis is labeled with 0, 1, and 2. The vertical axis is labeled with -2, 1, and 3. The origin is marked with a small circle. The axes are drawn with blue ink.

Hand-drawn diagram of a 2D array with 2 rows and 2 columns. The top row contains the values 4 and 7. The bottom row contains the values 3 and 4. A circle is drawn around the value 4 in the bottom-right cell. Below the array, the text "10 hi" is written.

1 2 4 2 6 3 8



6 8 7



≤ 5



> 5