BUBBLE SORT

$(\underset{0}{13}, \underset{1}{46}, \underset{2}{24}, \underset{3}{52}, \underset{4}{20}, \underset{5}{9})$

// push the max to the last

// compare adjacent member swap if
(left > right)

// the repeat for size less 1 than previous

this can be i

$$\text{first loop} \rightarrow 0 \rightarrow \overline{(n-1)}$$
$$0 \rightarrow (n-2)$$
$$0 \rightarrow (n-3)$$
$$\vdots$$
$$0 \rightarrow 1$$

as last element won't have anyone after to compare with

```
for ( i=n-1 ;  i≥0 ; i-- ){
    for ( j=0 ;  j<= i-① ;  j++ ){
        if (arr[j] > arr[j+1]) swap;

    }

}
```

COMPLEXITY $\rightarrow (n) + (n-1) + (n-2) + (n-3) \cdots + 1$

$$= \frac{n(n+1)}{2} = O(n^2) \rightarrow \text{worst}$$

average

```
        }
    ~~int temp =~~ a
    if (minIndex != i){
         int temp = arr[minIndex];
         arr[minIndex] = arr[i];
         arr[i]= temp;

    }

  }

}
```

COMPLEXITY -

inner loop → $\overset{\text{1st}}{\overset{\downarrow}{n}}, \overset{\text{2nd}}{\overset{\downarrow}{n-1}}, n-2, n-3, \ldots 2$ -

runs

5, 4, 3, 2 ...

approx to sum of natural numbers

$$= \frac{n(n+1)}{2} \Rightarrow \frac{n^2}{\cancel{2}} + \cancel{\frac{n}{2}}$$

$O(n^2)$

Best worst average

SELECTION SORT

13	46	24	52	20	9
0	1	2	3	4	5

// pick minimum (9)

// place at first position by swapping

// pick min → swap → with second index

. . . . .

step →	9	46	24	52	20	15
step 2 →	9	13	24	52	20	46
step 3 →	9	13	20	52	24	46
step 4 →	9	13	20	24	52	46
step 5 →	9	13	20	24	46	52

```
void selectionSort (int arr[]) {
    int size = arr.length;
    ~~int minIndex = i;~~          → going upto (n-2)
    for (int i=0; i<size-1; i++){
        int minIndex=i;          ^ (n-1)
        for (int j=i+1; j<size; j++){
            if (arr[j] < arr[minIndex]) {
                minIndex = j
}
```