```
BUBBLE.SORT (int[] array) {
    int last = array.length - 1;
    WHILE (last > 0) {
        int lastSwap = 0;
        int i = 0;
        WHILE (i < last) {
            if (array[i] > array[i+1]) {
                lastSwap = i;
                int temp = array[i];
                array[i] = array[i+1];
                array[i+1] = temp;
            }
            i++;
        }
        last = lastSwap;
    }
}
```

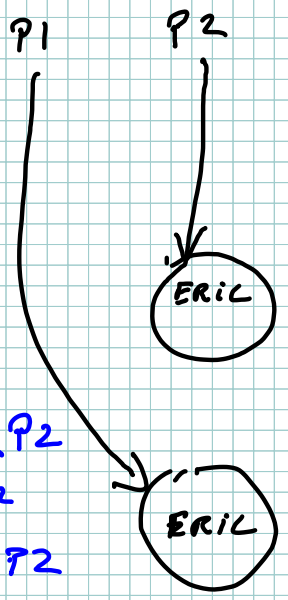
if(array[i].compareTo(array[i+1])>0)

comparing Objects;

Person p1, p2;

p1 == p2 ✗

p1.equals(p2) → true / false



Comparable Interface

p1.compareTo(p2) → negative (p1 < p2) / 0 (p1 == p2) / Positive (p1 > p2)

Integer i1, i2

i1.compareTo(i2) → -1 (i1 < i2) / 0 (i1 == i2) / +1 (i1 > i2)

```
selectionSort (int[] array) {
    int cur, min;
    for(int cur = 0 ; cur < array.length ; cur++){
        min = cur;
        for(int j = cur+1; j < array.length; j++) {
            if (array[j] < array[min]) {
                min = j;
            }
        }
        if(min != cur) {
            temp = array[min];
            array[min] = array[cur];
            array[cur] = temp
        }
    }
}
```

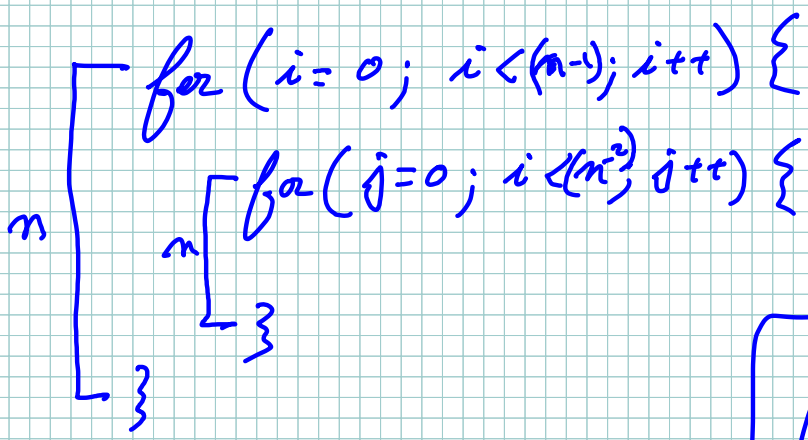
```
insertionSort(int[] array) {
    for (i = 1; i < array.length; i++) {
        temp = array[i];
        j = i;
        while ((j > 0) && (array[j-1] > temp)) {
            array[j] = array[j-1];
            j--;
        }
        array[j] = temp;
    }
}
```

temp = 55

0	1	2	3	4	5	6	7	8	9
8	10	15	18	22	33	42	55	72	75

$O(n^2)$

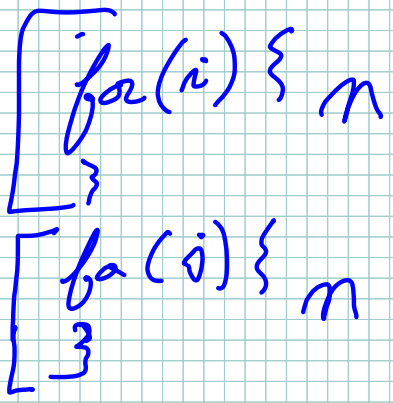
$O(n \log n)$



$(n * n)$

$O(n^2)$

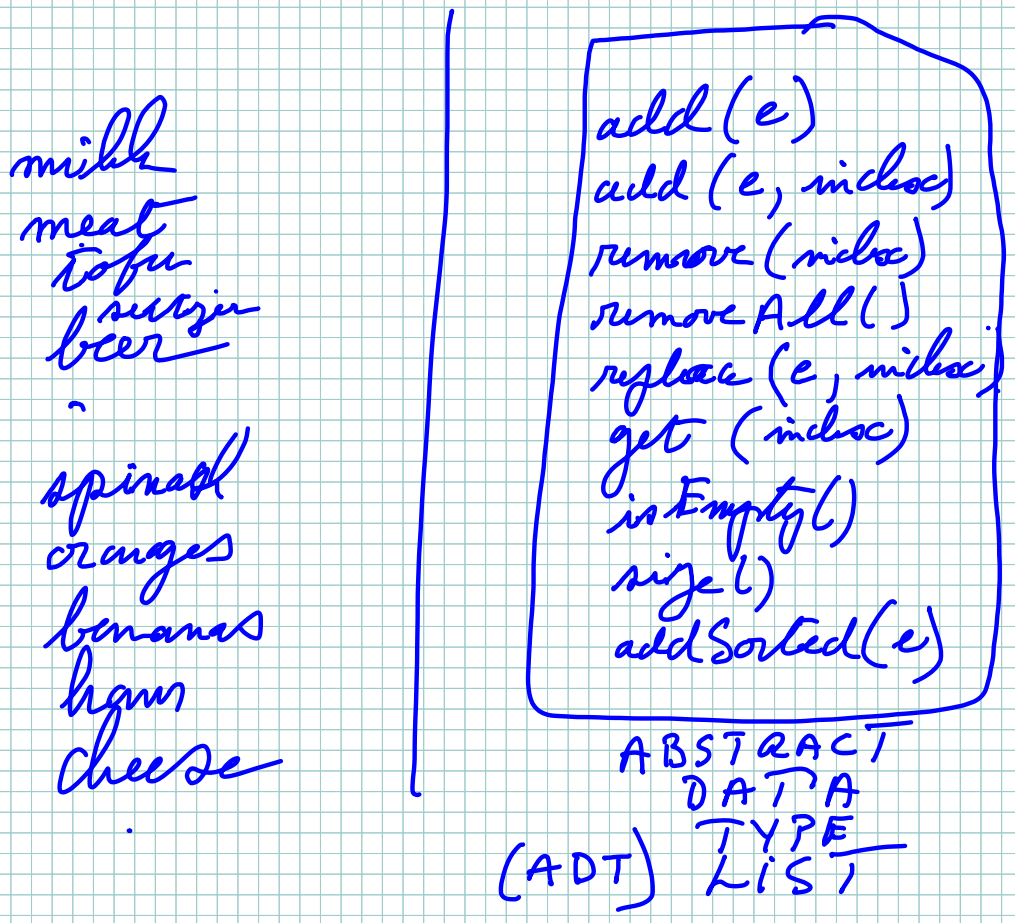
$O(n^2)$



$(n + n)$

$(2n)$

$O(n)$


milk
meat
tofu
sausage
beer
spinach
oranges
bananas
ham
cheese
add(e)
add(e, index)
remove(index)
removeAll()
replace(e, index)
get(index)
isEmpty()
size()
addSorted(e)
ABSTRACT DATA TYPE (ADT) LIST