

```

TestTimes runLinearSearch (int[] array, int target,
                           int num) {
    TestTimes tt = new TestTimes();
    LinearSearch ls = new LinearSearch();
    for (i=0 ; i < num ; i++) {
        long start = System.nanoTime();
        → int index = ls.search(array, target);
        long end = System.nanoTime();
        long testTime = end - start;
        tt.addTestTime(testTime);
    }
    return tt;
}

```

```

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;
        }
    }
}

```

$O(n^2)$
array sorted
 $O(n)$

n^2 Swaps
max

}

/

,

int i, j

$i > j$

$i == j$

$i < j$

Person P_1, P_2, P_3

JAMES
20

$P_1 > P_2$

$P_1 == P_2$

$P_1 == P_3$ ✓

KATHY
21

$P_1.equals(P_2)$

$P_1.compareTo(P_2)$

$P_2.compareTo(P_1)$

negative

positive

> <

$P_2.compareTo(P_3)$ ZERO

public class Person implements Comparable {

int compareTo(Object o) {

}

$P_1.compareTo(P_2)$

< 0
 $P_1 < P_2$
NEGATIVE

0
 $P_1 = P_2$

> 0
 $P_1 > P_2$
POSITIVE

}

```

selection Sort (int[] array) {
    int cur, min;
    for (cur = 0; cur < array.length; cur++) {
        min = cur;
        for (i = cur; i < array.length; i++) {
            if (array[i] < array[min]) {
                min = i;
            }
        }
        if (min != cur) {
            temp = array[min];
            array[min] = array[cur];
            array[cur] = temp;
        }
    }
}

```

$O(n^2)$
sorted array
 $O(n^2)$

n swap
max