

```

TestTimes run (arr, t, num) {
    TestTimes it = new TestTimes();
    for (i = 0; i < num; i++) {
        long startTime = System.nanoTime();
        int index = linearSearch(arr, t);
        long endTime = System.nanoTime();
        long runtime = endTime - startTime;
        it.addTestTime(runtime);
    }
    return it;
}

```

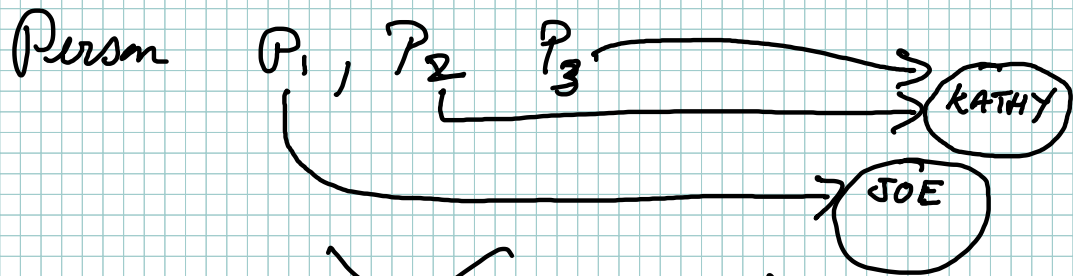
```

bubbleSort (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 already
 Sorted
 $O(n)$

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



~~$P_1 = P_2$~~

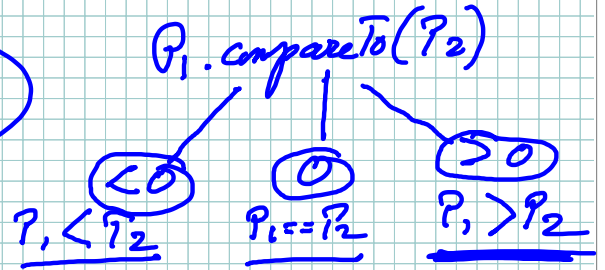
~~$P_1 = P_3$~~

public class Person implements Comparable {

```
int compareTo(Object o) {
```

}

}



```

Selection Sort (int[] array) {
    int cur, min;
    for (cur = 0; cur < array.length; cur++) {
        min = cur;
        for (i = cur + 1; i < array.length; i++) {
            if (array[i] < array[min]) {
                min = i;
            }
        }
        if (min != cur) {
            int temp = array[min];
            array[min] = array[cur];
            array[cur] = temp;
        }
    }
}

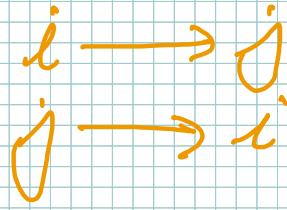
```

$O(n^2)$

Diagram annotations:

- A large blue bracket on the left side of the code spans from the start of the first for loop to the end of the second for loop, labeled with n .
- A smaller blue bracket on the left side of the code spans from the start of the inner for loop to the end of the inner for loop, labeled with n .
- A green circle highlights the swap logic block (the if statement and its contents), labeled with n .
- A yellow line underlines the comparison `array[i] < array[min]` in the inner loop.
- A yellow line underlines the swap logic block.
- Orange text at the bottom shows an alternative comparison: `if (array[i].compareTo(array[min]) < 0)`.

int i, j, temp



temp = i
i = j
j = temp