



BUBBLE SORT

$O(n^2)$

```

last = array.length - 1;
while (last > 0) {
  i = 0; lastSwap = 0;
  while (i < last) {
    if (array[i] > array[i+1]) {
      temp = array[i];
      array[i] = array[i+1];
      array[i+1] = temp;
      lastSwap = i;
    }
    i++;
  }
  last = lastSwap;
}

```

Selection Sort

```
cur = 0;  
min = 0;
```

```
→ 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) {  
        temp = array[min];  
        array[min] = array[cur];  
        array[cur] = temp;  
    }  
}
```

$O(n^2)$
selection Sort
has less
swaps than
bubble Sort

SWAP

Integer $i_1, i_2;$

$i_1.compareTo(i_2)$

Person $P_1, P_2;$

~~$P_1 < P_2$~~

Comparable Interface

int compareTo(Object o)

↑

$P_1.compareTo(P_2)$

{
-1 $P_1 < P_2$
0 $P_1 == P_2$
1 $P_1 > P_2$

Insertion Sort

```
int j, temp;
```

```
for (i=1; i < array.length; i++) {
```

```
    temp = array[i];
```

```
    j = i;
```

```
    while ((j > 0) && (array[j-1] > array[i])) {
```

```
        array[j] = array[j-1];
```

```
        j--;
```

```
    }
```

```
    array[j] = temp;
```

```
}
```

$O(n^2)$

1