

Bubble Sort Algorithm

```
for (int outer=0; outer<a.length-1;
    outer++) {
    for (int inner=0;
        inner<a.length-1-outer;
        inner++) {
        if (a[inner+1] < a[inner]) {
            int tmp = a[inner];
            a[inner] = a[inner+1];
            a[inner+1] = tmp;
        }
    }
}
```

Bubble Sort

Number of comparisons

$$n(n-1) / 2$$

Number of swaps

average case

$$n(n-1) / 4$$

worst case

$$n(n-1) / 2$$

Selection Sort Algorithm

```
for (int outer = 0; outer < a.length-1;
    outer++) {
    int min = outer;
    for (int inner=outer+1;
        inner <a.length; inner++) {
        if (a[inner] < a[min])
            min = inner;
    }
    int temp = a[outer];
    a[outer] = a[min];
    a[min] = temp;
}}
```

Selection Sort

Number of comparisons $n(n-1) / 2$

Number of swaps

average case

$n/2$

worst case

n

Insertion Sort Algorithm

```
for(int outer=1; outer<a.length; outer++)
{
    temp = a[outer];
    int inner=outer;
    for (; ((inner>0)&&(a[inner-1]>temp))
           inner--)
        {
            a[inner] = a[inner-1];
        }
    a[inner] = temp;
}}
```

Insertion Sort

Number of comparisons

worse case

$$n(n-1)/2$$

average case

$$n(n-1)/4$$

Number of copies/shifts, same as comparison