

Selection Sort

```
public static void selectionSort( Comparable [ ] a )
{
    for( int p = 0; p < a.length-1; p++ )
    {
        int minIndex = p;
        for( j = p+1; j < a.length-1; j++ )
            if ( a[minIndex].compareTo( a[ j ] ) > 0 )
                minIndex = j;

        Comparable tmp = a[ p ];
        a[p] = a[minIndex];
        a[minIndex] = tmp;
    }
}
```

Figure 8.3

Basic action of insertion sort (the shaded part is sorted)

Array Position	0	1	2	3	4	5
Initial State	8	5	9	2	6	3
After $a[0..1]$ is sorted	5	8	9	2	6	3
After $a[0..2]$ is sorted	5	8	9	2	6	3
After $a[0..3]$ is sorted	2	5	8	9	6	3
After $a[0..4]$ is sorted	2	5	6	8	9	3
After $a[0..5]$ is sorted	2	3	5	6	8	9

Figure 8.4

A closer look at the action of insertion sort (the dark shading indicates the sorted area; the light shading is where the new element was placed).

Array Position	0	1	2	3	4	5
Initial State	8	5				
After a[0..1] is sorted	5	8	9			
After a[0..2] is sorted	5	8	9	2		
After a[0..3] is sorted	2	5	8	9	6	
After a[0..4] is sorted	2	5	6	8	9	3
After a[0..5] is sorted	2	3	5	6	8	9

Insertion Sort

```
public static void insertionSort( Comparable [ ] a )
{
    for( int p = 1; p < a.length; p++ )
    {
        Comparable tmp = a[ p ];
        int j = p;

        for( ; j > 0 && tmp.compareTo( a[ j - 1 ] ) < 0; j-- )
            a[ j ] = a[ j - 1 ];
        a[ j ] = tmp;
    }
}
```

Figure 8.5

Shellsort after each pass if the increment sequence is {1, 3, 5}

ORIGINAL	81	94	11	96	12	35	17	95	28	58	41	75	15
After 5-sort	35	17	11	28	12	41	75	15	96	58	81	94	95
After 3-sort	28	12	11	35	15	41	58	17	94	75	81	96	95
After 1-sort	11	12	15	17	28	35	41	58	75	81	94	95	96

ShellSort

```
public static void shellsort( Comparable [ ] a )
{
    for( int gap = a.length / 2; gap > 0;
        gap = gap == 2 ? 1 : (int) ( gap / 2.2 ) )
        for( int i = gap; i < a.length; i++ )
        {
            Comparable tmp = a[ i ];
            int j = i;

            for( ; j >= gap && tmp.compareTo( a[ j - gap ] ) < 0; j -= gap )
                a[ j ] = a[ j - gap ];
            a[ j ] = tmp;
        }
}
```