Step Into Java:
Sorting Algorithms

Mr. Neat

Java

Suppose we have an array of something and we want it sorted in order.
How do we do it?

Get the biggest # to the right

57	95	88	14	25	6
57	88	95	14	25	6
57	88	14	95	25	6
57	88	14	25	95	6
57	88	14	25	6	95

(1st pass)

Get the biggest # to the right

57	88	14	25	6	95
57	88	14	25	6	95
57	14	88	25	6	95
57	14	25	88	6	95
57	14	25	6	88	95

(2nd pass)

Get the biggest # to the right

57	14	25	6	88	95
14	57	25	6	88	95
14	25	57	6	88	95
14	25	6	57	88	95

(3rd pass)

Get the biggest # to the right

14	25	6	57	88	95
14	25	6	57	88	95
14	6	25	57	88	95

(4th pass)

Get the biggest # to the right

14 6 25 57 88 95

6 14 25 57 88 95

(5th pass)

Get the biggest # to the right

57	95	88	14	25	6
57	88	14	25	6	95
57	14	25	6	88	95
14	25	6	57	88	95
14	6	25	57	88	95
6	14	25	57	88	95

(summary)

```
public static void sort(int[] list)
{
    for(int outer = 0; outer <list.length-1; outer++)
    {
        for(int inner =0; inner<list.length-outer-1; inner++)
        {
            if(list[inner] > list[inner+1])
            {
                //swap list[inner] and list[inner+1]
                int temp = list[inner];
                list[inner] = list[inner+1];
                list[inner+1] = temp;
            }
        }
    }
}
```

Why is this a void method?

```
public static void sort(int[] list)
{
    for(int outer = 0; outer <list.length-1; outer++)
    {
        for(int inner =0; inner<list.length-outer-1; inner+
        {
            if(list[inner] > list[inner+1])
            {
                //swap list[inner] and list[inner+1]
                int temp = list[inner];
                list[inner] = list[inner+1];
                list[inner+1] = temp;
            }
        }
    }
}
```

Get the biggest # to the right

outer	57	95	88	14	25	6
0	57	88	14	25	6	95
1	57	14	25	6	88	95
2	14	25	6	57	88	95
3	14	6	25	57	88	95
4	6	14	25	57	88	95

(summary)