Better Sort

Tuesday, March 17, 2020 1:38 PM

size = 5 , i = 0, j = 0

```
for(i=0; i<size; i++){
        for(j=i+1; j<size; j++){
            if (array[i]>array[j]){
                temp= array[i];
                array[i]= array[j];
                array[j]=temp;
            }
        }
    }
```

<- will run 5 times

<- sets up a subarray loop - the first element will become lowest. The i+1 for j truncates the space left to sort

we don't want to compare an element in the array to itself. In the first loop, we are comparing array [0] to [1] we then compare array [0] to the rest of the elements in the array. If something is bigger than array [0], we swap out the value that was in array [0] and replace it with the bigger value from array [j]. Our old big value moves to array [j]. Then we continue the loop to see if anything else was bigger.

When we finish the inner loop, i becomes 1 array [0] is the maximum / biggest value, we don't need to check it again. j = i+1, which will start at 2) the explenation here will repeat for array [1] comparing to array [2] through array [4]