Today, we will develop some of the algorithms that we mentioned last time. Then, we will move on to some of the algorithms shown in this handout. These slides will be used next time, too. We will attempt to go slowly enough that you can all understand the concepts and learn to develop these algorithms.

Do exercises on pp. 311–312 to get familiar with concepts and develop skill. Practice in DrJava! Test your methods!

**Sorting:**

```
Sorting: 0 n        "sorted" means in ascending order
pre: b  ? post: b  sorted  n
insertion sort 0 i sorted  ?
i
for (int i = 0; i < n; i = i + 1) {
    Push b[i] down into its sorted position in b[0..i];
}
Iteration i makes up to i swaps.
In worst case, number of swaps needed is
0 + 1 + 2 + 3 + … (n-1) = (n-1)*n / 2.
Called an “n-squared”, or n², algorithm.
```

**QuickSort**

```
/** Sort b[h..k] */
public static void qsort(int[] b, int h, int k) {
    if (b[h..k] has fewer than 2 elements) return;
    int j = partition(b, h, k);
    // b[h..j–1] <= b[j] <= b[j+1..k]
    // Sort b[h..j–1] and b[j+1..k]
    qsort(b, h, j–1);
    qsort(b, j+1, k);
}
```

Tony Hoare, in 1968
Quicksort author

Thought of Quicksort in ~1958. Tried to explain it to a colleague, but couldn’t. Few months later: he saw a draft of the definition of the language Algol 58—later turned into Algol 60. It had recursion. He went and explained Quicksort to his colleague, using recursion, who now understood it.

**Comments on A5**

Recursion:
Make requirements/descriptions less ambiguous, clearer; give more direction.
Need optional problem with more complicated recursive solution would have been an interesting challenge, more recursive functions. They make us think!
Make task 5 easier. I could not finish it.

```
I had intended here to erupt in largely incoherent rage over that wretched concept of recursion, which I came to hate like an enemy: like a sentient being who, knowing the difference between right and wrong, had purposely chosen to do me harm.
However, I then figured out how it works, and it is actually quite elegant, so now I suppose I have learned something against my will.
```

```
Good time drinking beer while watching the demo after I was done.
```

Tony Hoare, in 2007
in Germany
I don’t like how we are forced to visualize things in Dr. Gries’ way. … Entire point of programming is to be able to look at things in different ways and come up with different solutions for one problem. Forcing us to think of things in his way and testing us on it has been detrimental to my learning because in my opinion it wastes time and confuses me. This course should focus more on solving problems rather than drawing folders to represent objects.

1. A model of execution of Java programs is needed in order to bring understanding.
2. Problem solving is the focus. The programs you wrote for A5, the algorithms we are now studying, and the way we develop them, could not have been possible without the basics that we have given you.
3. We are giving you tools for coming up with good solutions, not just different ones.

1. Mark Twain: Nothing needs changing so much as the habits of others.