Chapter 13 presents several common algorithms for sorting an array of integers.

Two slow but simple algorithms are Selectionsort and Insertionsort.

This presentation demonstrates how the two algorithms work.
Sorting an Array of Integers

The picture shows an array of six integers that we want to sort from smallest to largest.
The Selectionsort Algorithm

- Start by finding the smallest entry.
The Selectionsort Algorithm

- Start by finding the smallest entry.
- Swap the smallest entry with the first entry.
The Selectionsort Algorithm

- Start by finding the **smallest** entry.
- Swap the smallest entry with the **first** entry.
Part of the array is now sorted.
The Selectionsort Algorithm

- Find the smallest element in the unsorted side.
The Selectionsort Algorithm

- Find the smallest element in the unsorted side.
- Swap with the front of the unsorted side.
We have increased the size of the sorted side by one element.
The Selectionsort Algorithm

- The process continues...

Bar graph showing the selection sort process with bars for [0], [1], [2], [3], [4], [5] and an arrow indicating the smallest element from the unsorted side.
The Selectionsort Algorithm

- The process continues...

Unsorted side

Swap with front
The Selectionsort Algorithm

- The process continues...

Sorted side is bigger

Unsorted side

[0] [1] [2] [3] [4] [5]
The Selectionsort Algorithm

- The process keeps adding one more number to the sorted side.
- The sorted side has the smallest numbers, arranged from small to large.
The Selectionsort Algorithm

- We can stop when the unsorted side has just one number, since that number must be the largest number.
The Selectionsort Algorithm

- The array is now sorted.
- We repeatedly selected the smallest element, and moved this element to the front of the unsorted side.
The Insertionsort Algorithm

- The Insertionsort algorithm also views the array as having a sorted side and an unsorted side.
The sorted side starts with just the first element, which is not necessarily the smallest element.
The Insertionsort Algorithm

- The sorted side grows by taking the front element from the unsorted side...
The Insertionsort Algorithm

- ...and inserting it in the place that keeps the sorted side arranged from small to large.
In this example, the new element goes in front of the element that was already in the sorted side.
Sometimes we are lucky and the new inserted item doesn't need to move at all.
The Insertionsort Algorithm

- Sometimes we are lucky twice in a row.
How to Insert One Element

1. Copy the new element to a separate location.
How to Insert One Element

2. Shift elements in the sorted side, creating an open space for the new element.
How to Insert One Element

2. Shift elements in the sorted side, creating an open space for the new element.
How to Insert One Element

2 Continue shifting elements...
How to Insert One Element

...until you reach the location for the new element.
How to Insert One Element

3 Copy the new element back into the array, at the correct location.
How to Insert One Element

- The last element must also be inserted. Start by copying it...
A Quiz

How many shifts will occur before we copy this element back into the array?
A Quiz

- Four items are shifted.
A Quiz

- Four items are shifted.
- And then the element is copied back into the array.
Both Selectionsort and Insertionsort have a worst-case time of $O(n^2)$, making them impractical for large arrays.

But they are easy to program, easy to debug.

Insertionsort also has good performance when the array is nearly sorted to begin with.

But more sophisticated sorting algorithms are needed when good performance is needed in all cases for large arrays.