Sorting

Why sort
- It’s nice to see data in sorted order
- Easier to search (binary search)
- Sorting used as a step in many algorithms

Sorting algorithms
- There are many algorithms for sorting:
  - Selection sort
  - Insertion sort
  - Bubble sort
  - Merge sort
  - Heap sort
  - Radix sort
  - Quick sort
  - Stooge sort
- Each has its advantages and disadvantages

Selection Sort
- Find the smallest item
- Put it in the first position
- Find the 2nd smallest item
- Put it in the 2nd position
- Find the 3rd smallest item
- Put it in the 3rd position
- ...

Selection Sort code
```python
def selection_sort(array):
    for i in range(len(array)):
        min = i;
        for j in range(i + 1, len(array)):
            if array[j] < array[min]:
                min = j;
        temp = array[i]
        array[i] = array[min]
        array[min] = temp
```
Insertion sort

- Works the same way you arrange your hand when playing cards.
- Pick up a card and place it in your hand in the correct position relative to the cards you're already holding.

Arranging a hand of cards

Insertion Sort

Insertion Sort (cont.)
Insertion Sort (cont.)

Insertion sort – more formally

- insertion sort partitions the array into two regions: sorted, and unsorted
- each iteration the sorted part grows by 1

Insertion sort – another example

Insertion Sort Algorithm

```python
def insertion_sort(array):
    for i in range(1, len(array)):
        temp = array[i]
        position = i
        # shift larger values to the right
        while (position > 0 and array[position-1] > temp):
            array[position] = array[position-1]
            position -= 1
        # insert the current item
        array[position] = temp;
```

Bubble Sort

- Compares neighboring elements, and swaps them if they are not in order
  - Effect: the largest value will "bubble" to the last position in the array.
  - Repeating the process will bubble the 2nd to largest value to the 2nd to last position in the array
Bubble Sort

def bubble_sort(array):
    for position in range(len(array)-1, 0, -1):
        for j in range(position):
            if array[i] > array[i+1]:
                temp = array[i]
                array[i] = array[i+1]
                array[i+1] = temp