# Sorting

Given a linear list of *comparable* objects of the same class (or values of the same type), we wish to sort (or reärrange) the objects in the increasing order.

For simplicity, let's just assume that we are given an array of *n* integers.



# Selection sort

- Find the least value in the array.
- Swap it with the value in the first index.
- Repeat the steps above for the remainder of the array, starting from the second index and advancing each time.

# Sorting



# Sorting

How should we do this?

One way: We know how to find the smallest number. We should be able to repeat this to find the smallest, the second smallest, the third smallest, etc.

But we also need to move these numbers to the correct locations.

# Bubble sort

- By comparing and (swapping if necessary) adjacent pairs of numbers, "bubble up" the largest value in the array.
- Repeat the above step for the remainder of the array, ending at the second-to-the last index and advancing each time.

# Selection sort

Suppose we are given an array `A` of `n` integers.

Pseudocode:

```
for (int i = 0; i < n - 1; i++) {
  int minIndex = i;
  for (int j = i + 1; j < n; j++)
    if (A[j] < A[minIndex])
      minIndex = j;
  if (minIndex != i)
    swap A[i] with A[minIndex];
}
```



# Bubble sort

Suppose we are given an array `A` of `n` integers.

```
for (int i = 1; i < n; i++)
  for (int j = 0; j < n - i; j++)
    if (A[j] > A[j + 1])
      swap A[j] with A[j + 1];
```



# "Swapping" Two Values

Suppose you want to swap two values, *m* and *n*.

Suppose *m = 1* and *n = 2*. Why won't this work? What values will *m* and *n* have after this "swap"?

```
m = n;
n = m;
```

# Swapping Two Values

Suppose you want to swap two values, *m* and *n*.

To swap *m* and *n*, you need a temporary variable:

```
temp = m;
m = n;
n = temp;
```