

Algorithms + Data Structures = Programs

# Algorithms in Java

- Until now we have looked at how to create data structures to represent objects in the world
- This week, we focus on algorithms, i.e. how to manipulate the data structures.

## Sorting an Array

- A simple algorithm to sort a fixed length array
  - Find smallest element
  - move it to the front
  - repeat until all the array has been scanned
  - simple but not the most efficient method

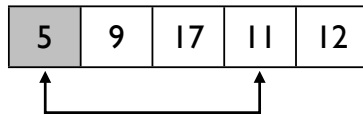
## Selection Sort

- Find the smallest element
- Swap it with the first element

11	9	17	5	12
----	---	----	---	----

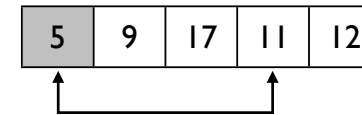
# Selection Sort

- Find the smallest element
- Swap it with the first element



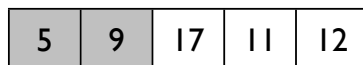
# Selection Sort

- Find minimum of remaining elements
- Swap it with the next element



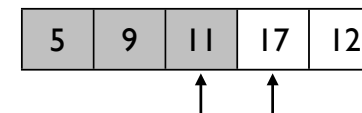
# Selection Sort

- Find minimum of remaining elements
- Swap it with the next element



# Selection Sort

- Repeat



# Selection Sort

- Repeat



## Sort Demo

### Main Method

```
public static void main()
{
    int[] a = randomIntArray(20, 100);

    print(a);           // print the unsorted array
    sort(a);           // sort the array
    print(a);           // print the sorted array
}
```

### Create a random array

```
public static int[] randomIntArray(int length, int n)
{
    int a[] = new int[length];
    Random generator = new Random();

    for (int i = 0; i < a.length; i++)
    {
        a[i] = generator.nextInt(n);
    }
    return a;
}
```

# Arrays and ArrayLists

- An **array** is a primitive type in Java
  - It is a fixed length collection of objects
  - Elements are always accessed by index
  - Size of array, A, is A.length;
- An **ArrayList** is a pre-defined class
  - Variable length, different access methods
  - Size of ArrayList, A, is A.size();

## Sort an array

```
public static void sort(int[] a)
{
    for (int n = 0; n < a.length - 1; n++)
    {
        int minPos = minimumPosition(a, n);

        if (minPos != n)
        {
            swap(a, minPos, n);
        }
    }
}
```

## Print an array

```
public static void print(int[] a)
{
    for (int i = 0; i < a.length; i++)
    {
        System.out.print(a[i] + " ");
    }
    System.out.println();
}
```

## Find the position of the smallest element

```
public static int minimumPosition(int[] a, int from)
{
    int minPos = from;

    for (int i = from + 1; i < a.length; i++)
    {
        if (a[i] < a[minPos])
        {
            minPos = i;
        }
    }
    return minPos;
}
```

# Swap two elements in an array

```
public static void swap(int[] a, int i, int j)
{
    int temp = a[i];
    a[i] = a[j];
    a[j] = temp;
}
```