Data Structures and Algorithm Analysis

Lab 3, algs4 and Union Find.

Contents

- Review how to use the algs4 library.
- Union Find.

Review: the algs4 library

The algs4 library is a JAVA library that helps you coding. It implements many commonly used functionalities.

algs4:

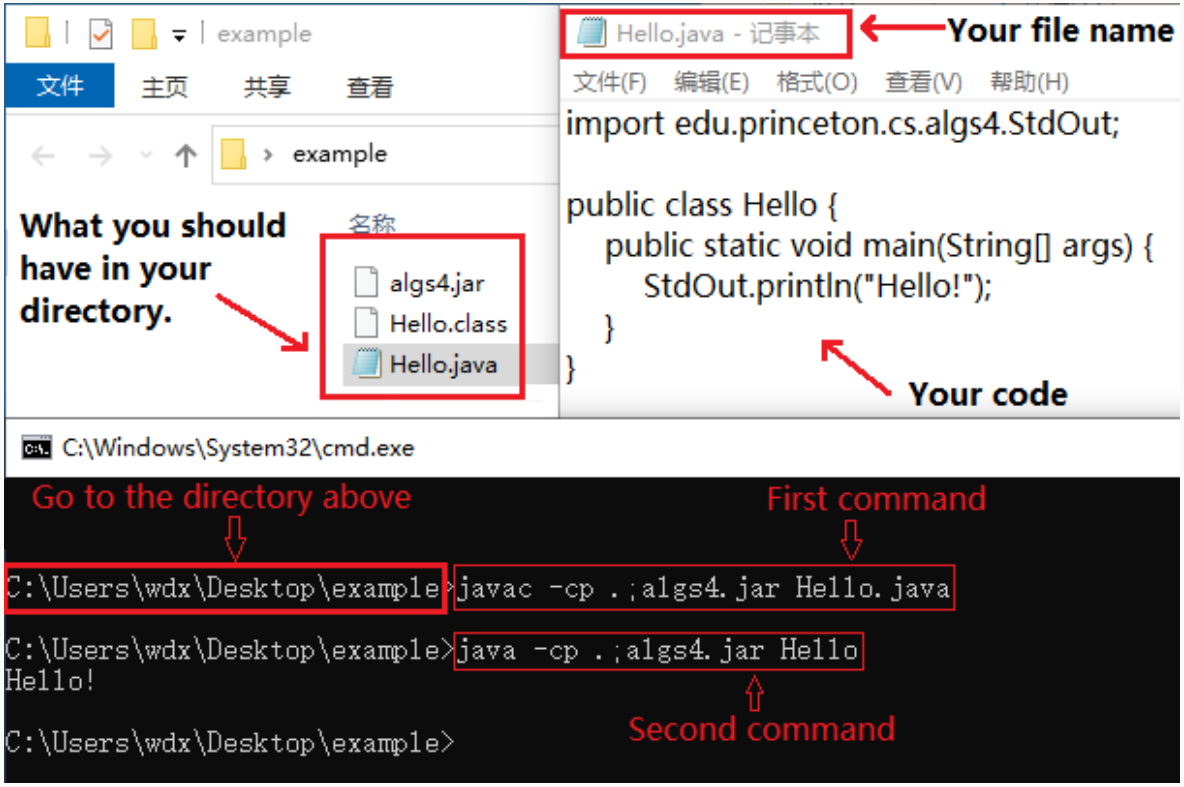
https://algs4.cs.princeton.edu/code/

Download:

https://algs4.cs.princeton.edu/code/algs4.jar

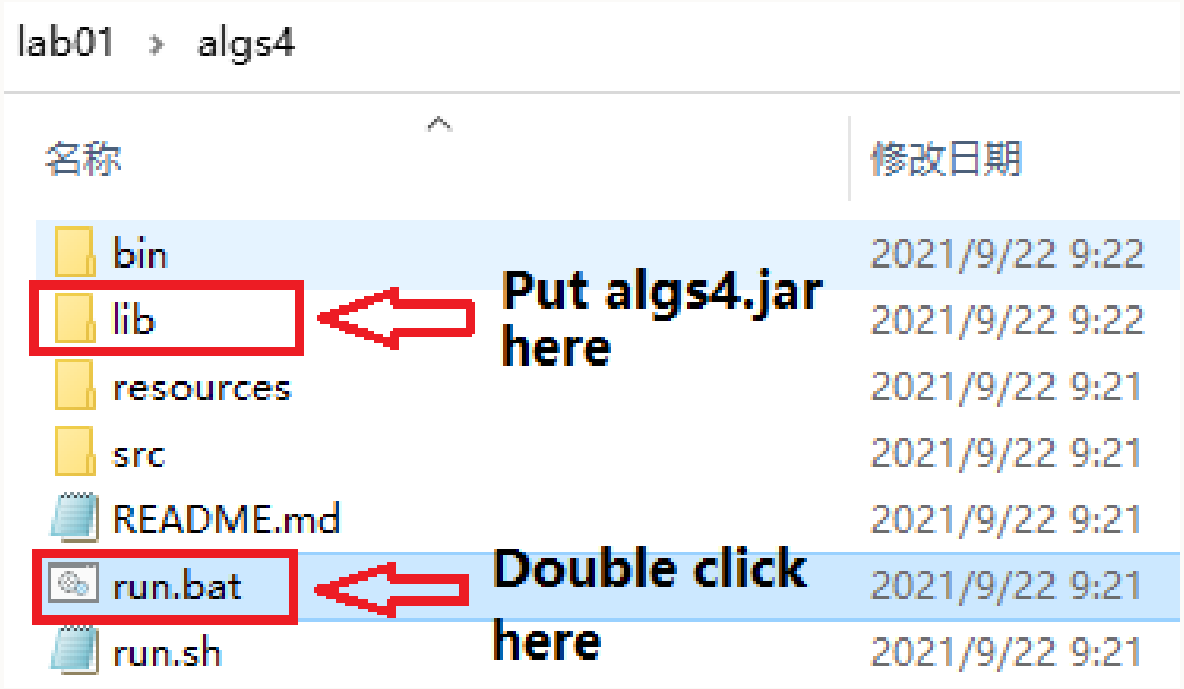
Using the algs4 in command line

Everything you need to know about how to use the algs4 library is in the following picture.



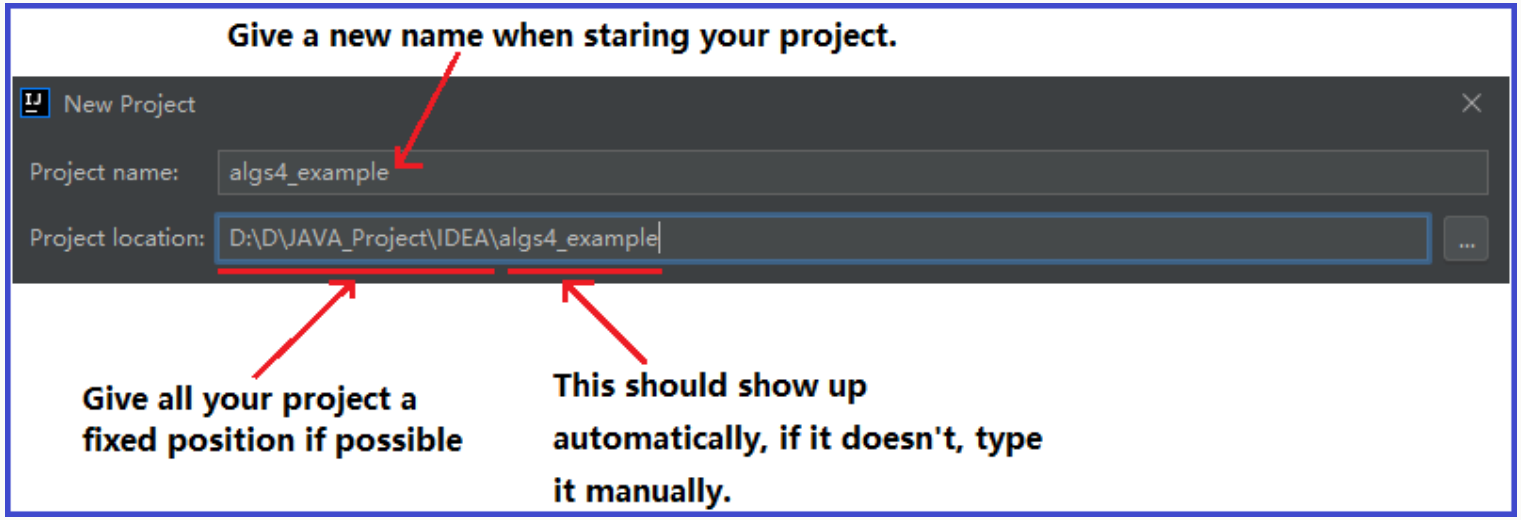
Using the algs4 in command line

If you still have any problem. Download the lab material in lab 1. It's an example project. Double click "run.bat" and it will run the examples. If you use linux or mac, run "run.sh".



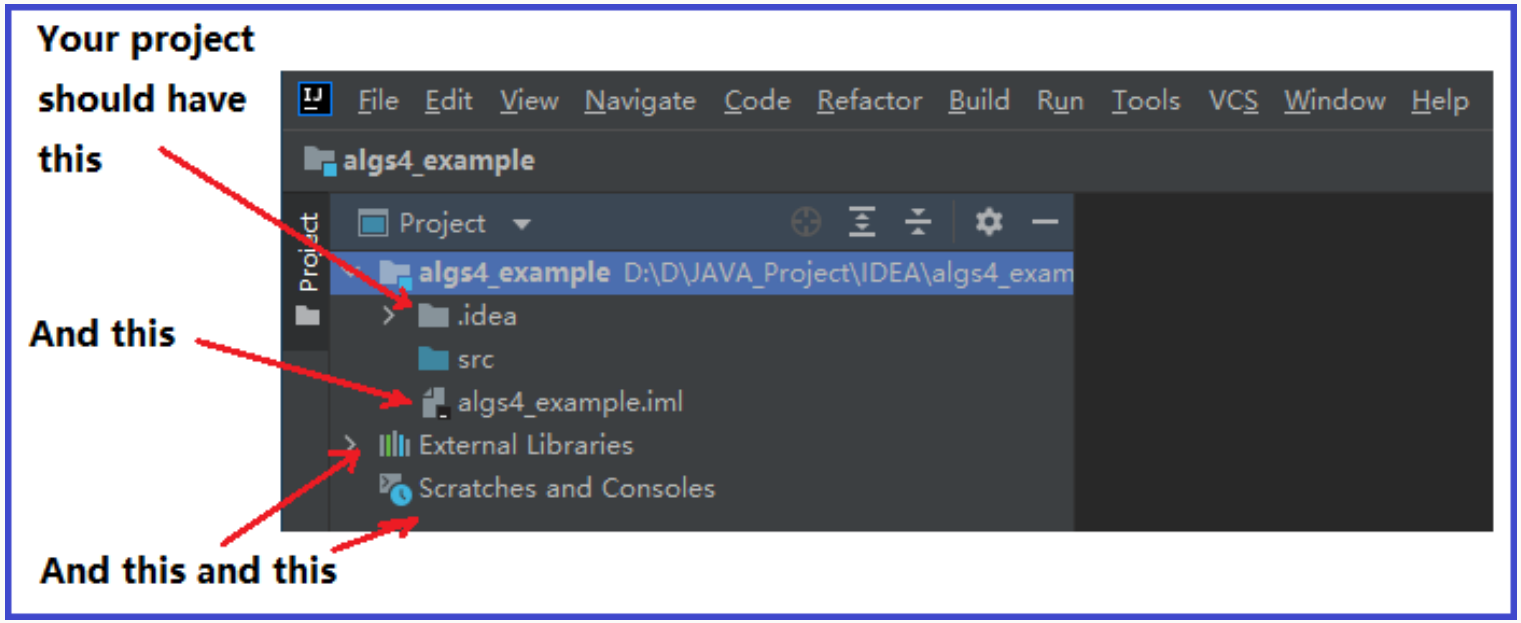
Using the algs4 in IntelliJ IDEA

If you use IDEA for your project management, you need to create a **new project**.



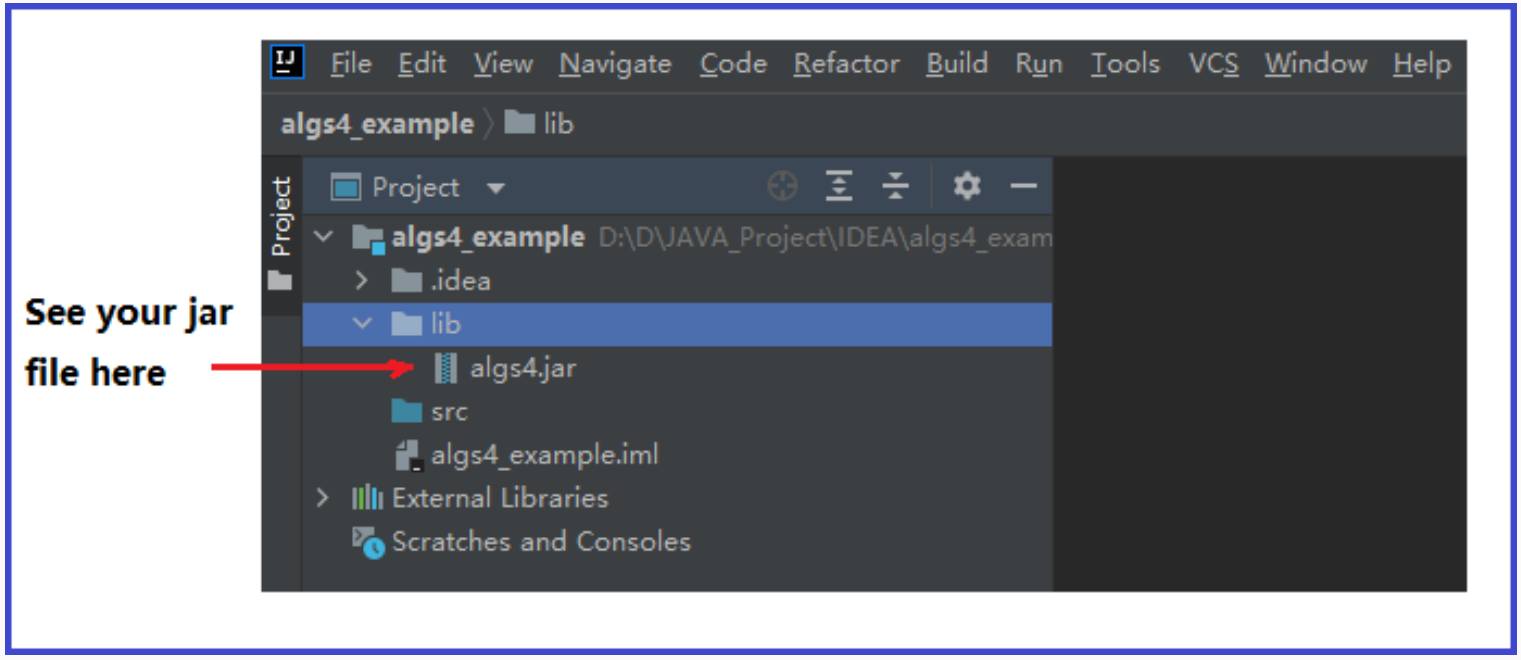
Using the algs4 in IntelliJ IDEA

You project should look like this:



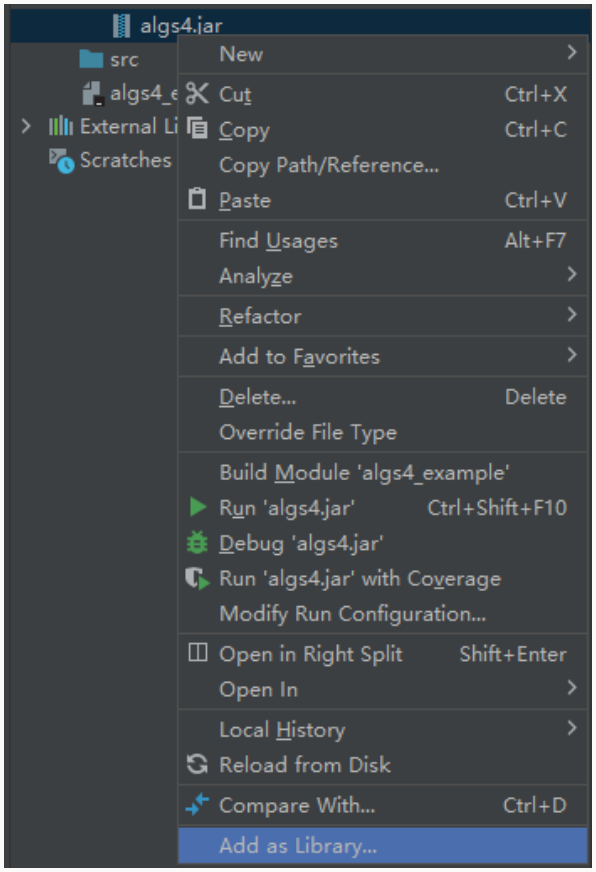
Using the algs4 in IntelliJ IDEA

Create a new folder named "lib", and copy the "algs4.jar" into "lib".



Using the algs4 in IntelliJ IDEA

Right click on your .jar file and select "add as library".



Using the algs4 in IntelliJ IDEA

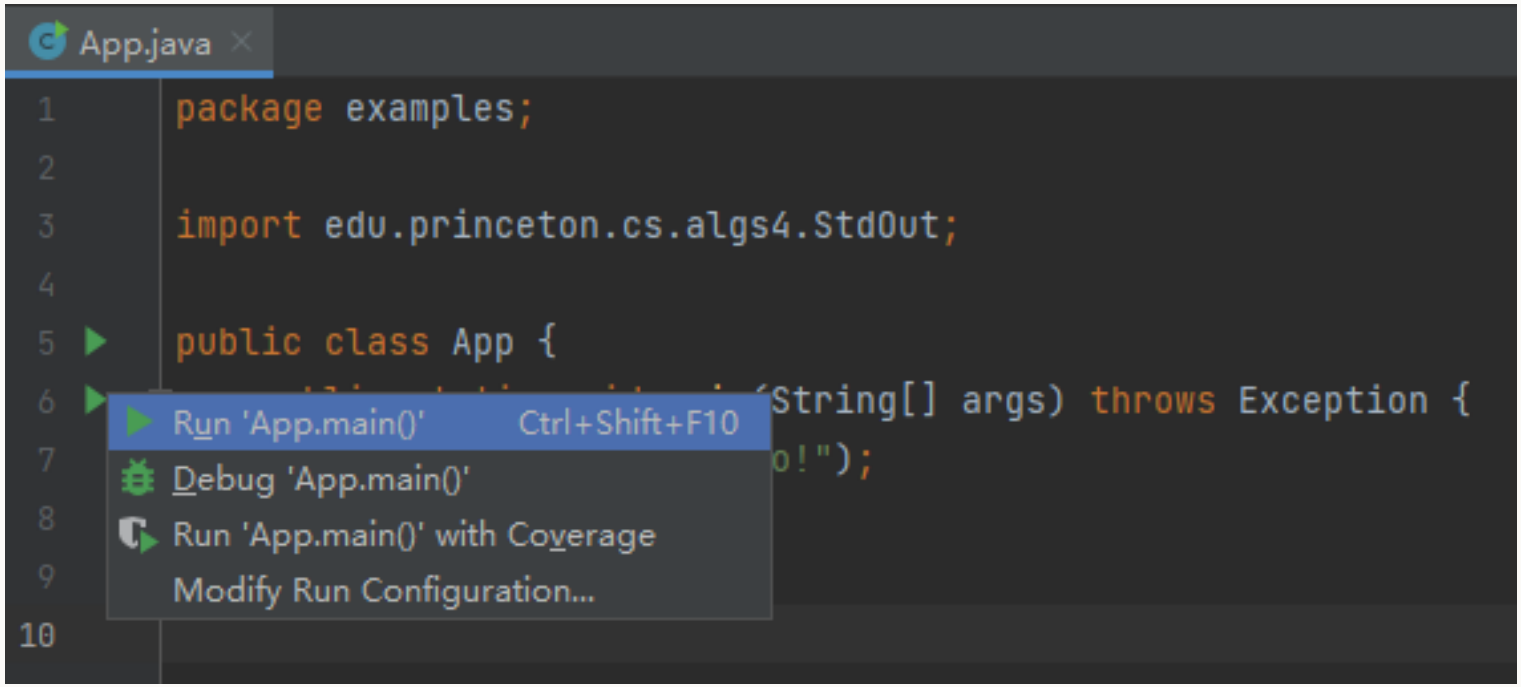
Then you can add your code in "src" directory.

App.java

```
1  package examples;
2
3  import edu.princeton.cs.algs4.StdOut;
4
5  public class App {
6      public static void main(String[] args) throws Exception {
7          StdOut.println("Hello!");
8      }
9  }
```

Using the algs4 in IntelliJ IDEA

Click the green triangle and run. It the triangles did not show up, you didn't create your project properly.



Using the algs4 in IntelliJ IDEA

Now you know how to use IDEA for your develop. Here are some advices:

- Do not copy and paste a project into another project. IDEA may not recognize it, and it may not run.
- Do not open a folder directly with IDEA. It may not be recognized as a project, and it may not run.
- Do not operate in the above way with the lab materials. They are **not** idea project. You need to **create a new project**.

Review: input and output with algs4

```
import edu.princeton.cs.algs4.StdIn;
import edu.princeton.cs.algs4.StdOut;

public class TestStdInStdOut {
  public static void main( String[] args ) {
    int i = StdIn.readInt();
    StdOut.print("Read int from input:");
    StdOut.println(i);

    String str = StdIn.readString();
    StdOut.print("Read String from input:");
    StdOut.println(str);

    String line = StdIn.readLine();
    StdOut.print("Read line from input:");
    StdOut.println(line);

    double[] darr = StdIn.readAllDoubles();
    StdOut.print("Read all doubles from input:");
    for( double d : darr )
      StdOut.print(d+" ");
    StdOut.println();
```

Review: input and output with arrays

```
import edu.princeton.cs.algs4.StdArrayIO;

public class TestStdArrayIO {
  public static void main( String[] args ) {
    int[] intArr = StdArrayIO.readInt1D();
    StdArrayIO.print(intArr);

    int[][] intArr2 = StdArrayIO.readInt2D();
    StdArrayIO.print(intArr2);

    double[][] douArr2 = StdArrayIO.readDouble2D();
    StdArrayIO.print(douArr2);

    boolean[][] boolArr2 = StdArrayIO.readBoolean2D();
    StdArrayIO.print(boolArr2);
  }
}
```

Review: binary search

This is the binary search we have used in the 3-sum problem.

```
import edu.princeton.cs.algs4.BinarySearch;
import edu.princeton.cs.algs4.StdOut;
import edu.princeton.cs.algs4.StdArrayIO;

public class TestBinarySearch {
  public static void main( String[] args ) {
    int[] arr = new int[] { 2, 3, 5, 7, 11, 13, 17, 19 };
    int target = 17;
    StdArrayIO.print(arr);
    StdOut.printf("Index of %d is %d.\n", target,
      BinarySearch.indexOf(arr, target));
  }
}
```

Insertion sort

The insertion sort you will be learning very soon.

```
import edu.princeton.cs.algs4.Insertion;
import edu.princeton.cs.algs4.StdOut;

public class TestInsertion {
  public static void main( String[] args ) {

    Integer[] arr = new Integer[] { 7, 11, 2, 5, 3, 19, 13
      };
    for( Integer i : arr )
      StdOut.print(" "+i);
    StdOut.println();

    Insertion.sort(arr);
    for( Integer i : arr )
      StdOut.print(" "+i);
    StdOut.println();
  }
}
```

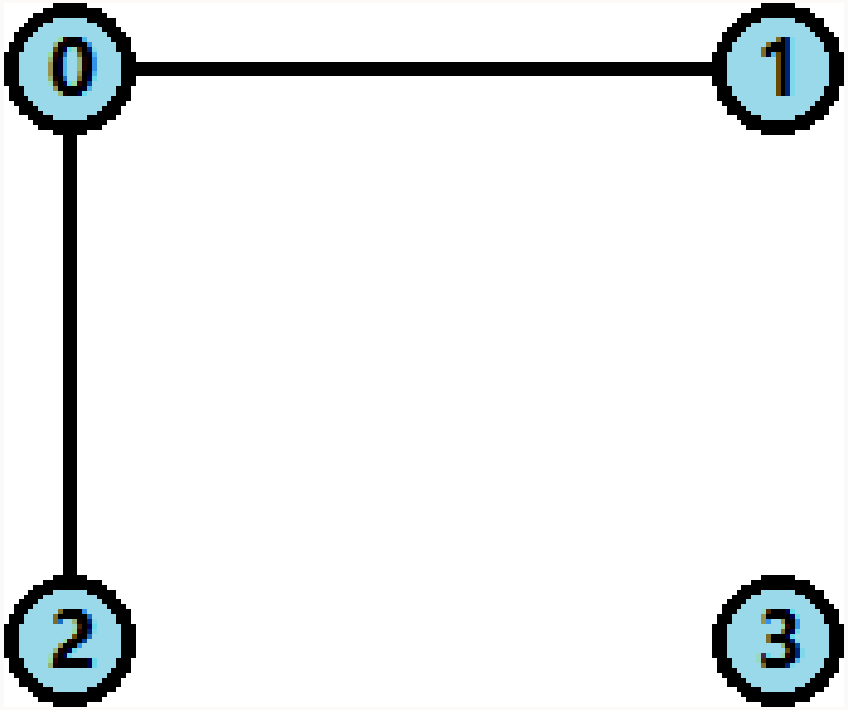
Merge sort

Merge sort uses a different way to sort the data.

```
import edu.princeton.cs.algs4.Insertion;
import edu.princeton.cs.algs4.StdOut;

public class TestInsertion {
  public static void main( String[] args ) {

    Integer[] arr = new Integer[] { 7, 11, 2, 5, 3, 19, 13
      };
    for( Integer i : arr )
      StdOut.print(" "+i);
    StdOut.println();

    Merge.sort(arr);
    for( Integer i : arr )
      StdOut.print(" "+i);
    StdOut.println();
  }
}
```

Undirected graph

Graphs are common in algorithm topics, this is a graph with 4 vertices.



Undirected graph

You can use the following code to see whether point 3 is reachable from point 0.

```
public static void main(String[] args) {
  Graph graph = new Graph(4);

  graph.addEdge(0, 1);
  graph.addEdge(1, 2);

  StdOut.println(graph);
}
```

Exercise 1: depth first search

Using depth first search to find out whether you can reach vertex "3" from vertex "0".

```
public static void main(String[] args) {
  Graph graph = new Graph(4);

  graph.addEdge(0, 1);
  graph.addEdge(1, 2);

  // Write your code here
}
```

Exercise 2: binary search

Implement binary search algorithm. It is required that the algorithm finds an interval and returns and array of two elements containing the index of the first and last element.

You don’t have to write everything. Just download lab 3 material and fill in the blanks in "BinaryRangeSearch.java".

Exercise 3: union find

You need to implement Union Find algorithm. You don't have to write everything. Just download lab 3 material and fill in the blanks in "UnionFind.java".

```
public class UnionFind {
  public boolean isConnected( int p, int q ) {
    // Write code here!
  }
  public void union( int p, int q ) {
    // Write code here!
  }
```