INTRODUCTION

Algorithms – method for solving a problem

Data structures – method to store information

Setting up IntelliJ

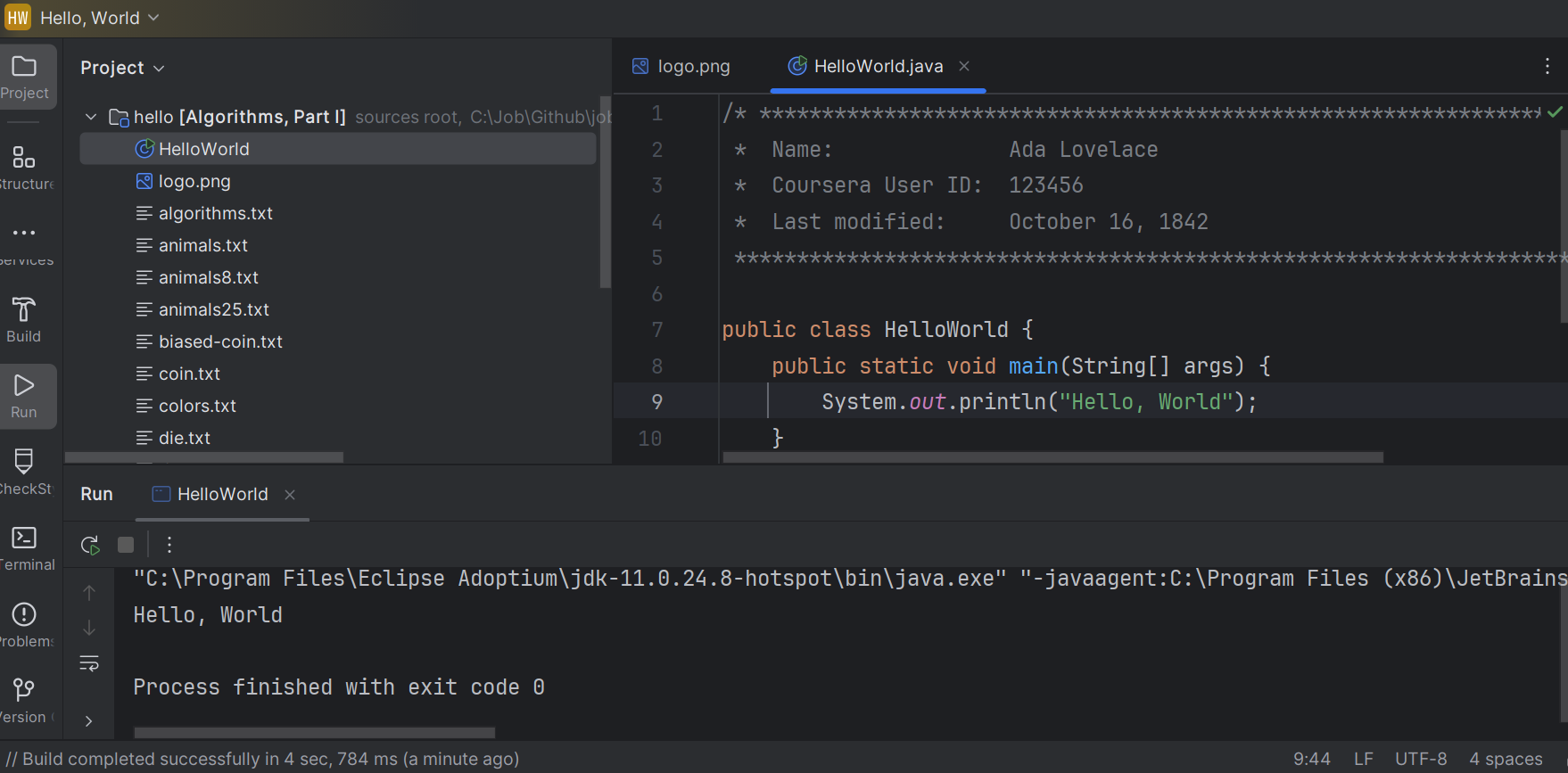
**LIFT → Project (Ctrl + 1)**

**LIFT → New Java Class**

**File → Save All (Ctrl + S)**. When you save the file, IntelliJ reformats your code (as necessary).

**LIFT → Recompile 'HelloWorld.java' (Ctrl + B)**.

To execute your program, select the menu option **LIFT → Run 'HelloWorld' with Arguments (Ctrl + E)**.



*command line* is

* **LIFT → Terminal (Alt + 2)**.

*compile* your program, type the following javac command. More specifically, type the text in yellow that appears on the same line as the command prompt.

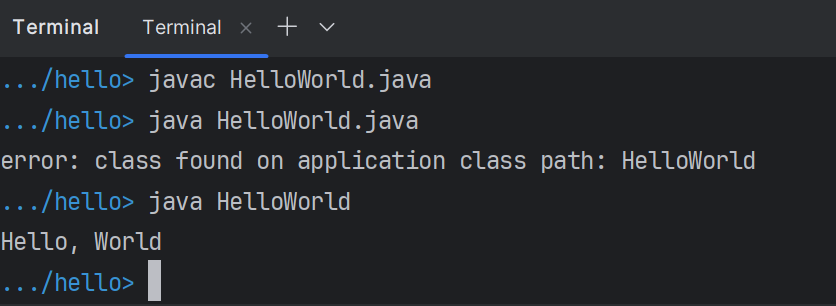
~> javac HelloWorld.java

~>

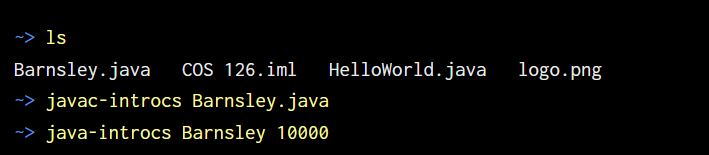
*execute* your program, type the following java command:

~> java HelloWorld

Hello, World



**ls** **command** to display the contents of the subdirectories recursively



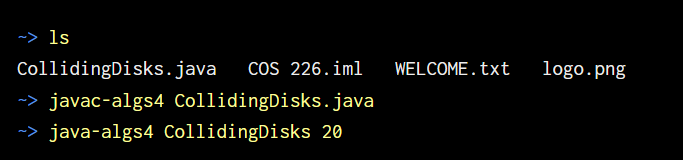
*Use the -classpath command-line option.* Put stdlib.jar in the same directory as the program you are writing (but do not unjar it). Then, compile and execute as follows:

|  |
| --- |
| OS X / Linux / Git Bash  -----------------------  % **javac -classpath .:stdlib.jar MyProgram.java**  % **java  -classpath .:stdlib.jar MyProgram**  Windows Commmand Prompt  -----------------------  % **javac -classpath .;stdlib.jar MyProgram.java**  % **java  -classpath .;stdlib.jar MyProgram** |

 *Use the CLASSPATH environment variable.* You can set your system CLASSPATH environment variable to contain stdlib.jar. We do not recommend this approach because the CLASSPATH variable may be used by other applications.

 *Configure your IDE.* You can configure your IDE to include stdlib.jar in the classpath.

* *IntelliJ*: select *File → Project Structure → Libraries*.
* *Eclipse*: select *Project → Properties → Java Build Path → Libaries → Add External JARs*.
* *DrJava*: select *Preferences → Extra Classpath → Add*.
* *Windows Command Prompt*: select *Start → Computer → System Properties → Advanced → Environment Variables → User Variables → CLASSPATH*.
* % **java Wget "https://introcs.cs.princeton.edu/java/data/codes.csv"**
* % **more codes.csv**
* United States,USA,00
* Alabama,AL,01
* Alaska,AK,02
* ...

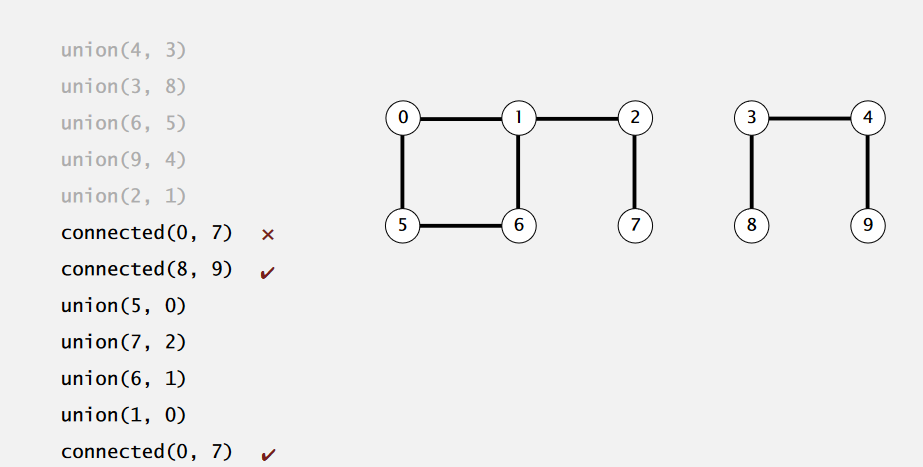


java-algs4 RandomWord < animals8.txt

UNION-FIND

Steps to develop a usable algorithm:

1. Model the problem
2. Find an algorithm to solve it
3. Fast enough? Fits in memory?
4. Figure out why
5. Find a way to address the problem
6. Iterate



We assume "is connected to" is an equivalence relation:

・Reflexive: p is connected to p.

・Symmetric: if p is connected to q, then q is connected to p.

・Transitive: if p is connected to q and q is connected to r,

then p is connected to r.

