

Data Structures and Algorithm Analysis

Lab 1, introduction of the algs4 library



Contents

- The algs4 library.
- Basic functionalities.

The algs4 library

We will use JAVA to implement and test all the algorithm in this course.

In order to ease the programming and focus on the algorithm, we will also use the algs4 library.

The algs4 library

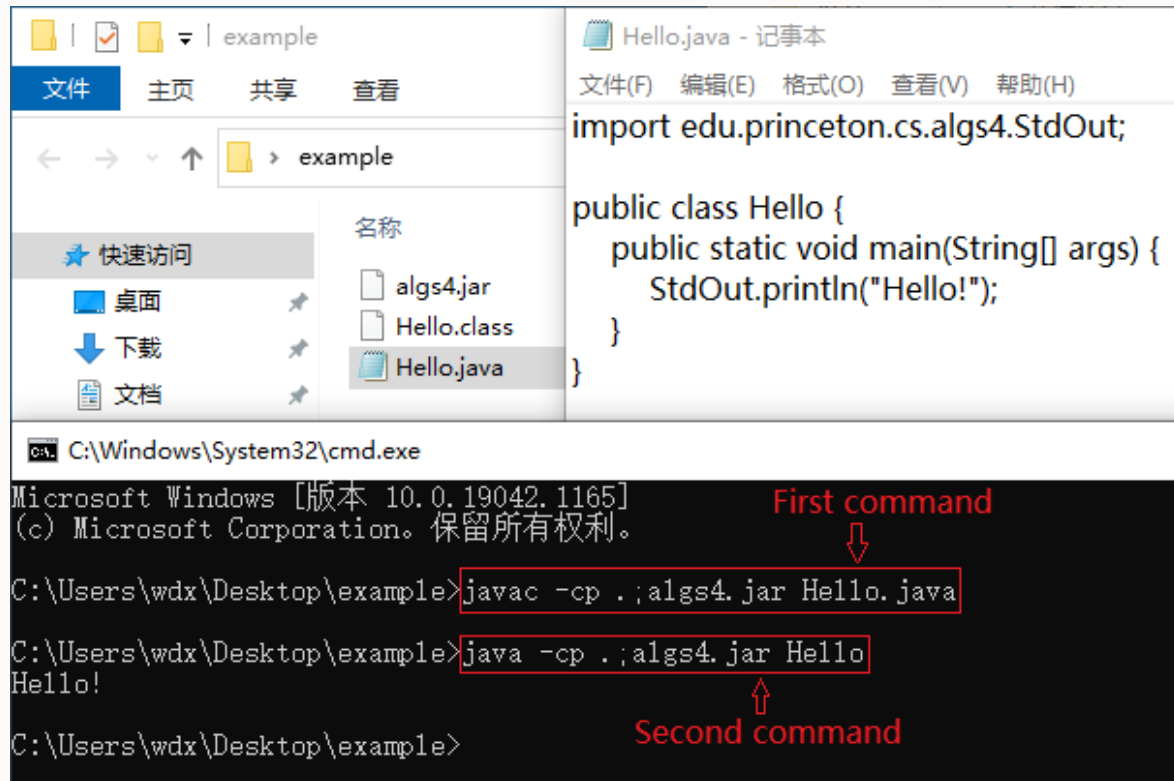
The algs4 library is a JAVA library designed to aid the learning of many commonly used algorithms.

Download at: <https://algs4.cs.princeton.edu/code/algs4.jar>

Library document at: <https://algs4.cs.princeton.edu/code/>

The algs4 library

Using algs4 is easy. Just put the algs4.jar in your classpath and import the functionality you want to use.



The screenshot shows a Windows file explorer window with the 'example' folder selected. The folder contains three files: 'algs4.jar', 'Hello.class', and 'Hello.java'. The 'Hello.java' file is selected. To the right, the code editor shows the following Java code:

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

public class Hello {
    public static void main(String[] args) {
        StdOut.println("Hello!");
    }
}
```

Below the file explorer, a command prompt window is open, showing the following commands and output:

```
C:\Windows\System32\cmd.exe
Microsoft Windows [版本 10.0.19042.1165]
(c) Microsoft Corporation。保留所有权利。

C:\Users\wdx\Desktop\example>javac -cp .;algs4.jar Hello.java

C:\Users\wdx\Desktop\example>java -cp .;algs4.jar Hello
Hello!

C:\Users\wdx\Desktop\example>
```

Red annotations highlight the commands:

- First command:** `javac -cp .;algs4.jar Hello.java`
- Second command:** `java -cp .;algs4.jar Hello`

Basic functionalities

The algs4 library contains many commonly used basic utilities. Implementing all these utilities all by yourselves could be time consuming and diverging from our goal.

- Writing to standard output.
- Reading from standard input.
- Draw simple shapes on a window.
- Generate random numbers from certainer
- Simple statistics of arrays.

Writing to standard output

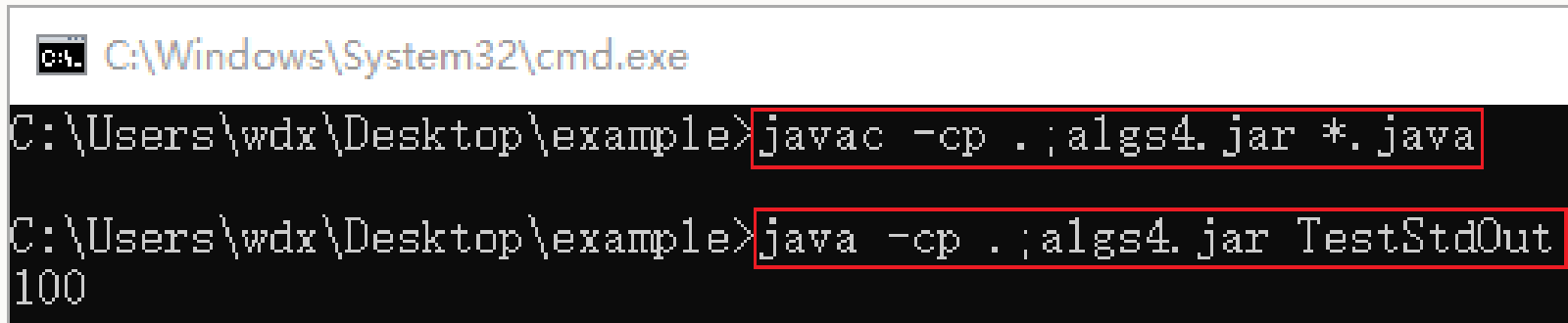
The StdOut class provides simple methods to write data to standard output. Its usage is similar to System.out.

Writing to standard output

Print an integer:

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

public class TestStdOut {
    public static void main( String[] args ) {
        int i = 100;
        StdOut.println(i);
    }
}
```



A screenshot of a Windows command prompt window. The title bar shows the icon for a command prompt and the path "C:\Windows\System32\cmd.exe". The command prompt shows the following commands and output:

```
C:\Users\wdx\Desktop\example>javac -cp .;algs4.jar *.java
C:\Users\wdx\Desktop\example>java -cp .;algs4.jar TestStdOut
100
```


Writing to standard output

Print a double:

```
import edu.princeton.cs.algs4.StdOut;  
  
public class TestStdOut {  
    public static void main( String[] args ) {  
        double d = 1.1;  
        StdOut.println(d);  
    }  
}
```

 C:\Windows\System32\cmd.exe

C:\Users\wdx\Desktop\example> javac -cp .;algs4.jar *.java

C:\Users\wdx\Desktop\example> java -cp .;algs4.jar TestStdOut
1.1

Writing to standard output

Print a string:

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

public class TestStdOut {
    public static void main( String[] args ) {
        String str = "Test String.";
        StdOut.println(str);
    }
}
```

 C:\Windows\System32\cmd.exe

```
C:\Users\wdx\Desktop\example>javac -cp .;algs4.jar *.java
```

```
C:\Users\wdx\Desktop\example>java -cp .;algs4.jar TestStdOut
Test String.
```

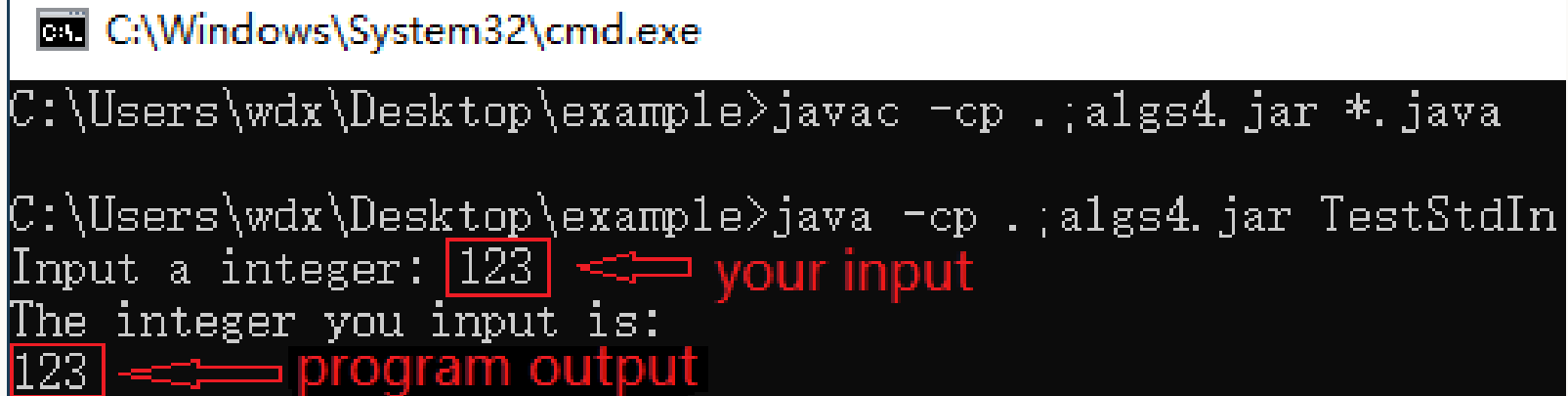
Reading from standard input

The `StdIn` class provides the methods to read from standard input. It internally uses a `Scanner` to provide the functionalities but unlike the `Scanner` class, it is used in a static way.

Reading from standard input

How to read a integer from standard input:


```
import edu.princeton.cs.algs4.StdOut;  
import edu.princeton.cs.algs4.StdIn;  
public class TestStdOut {  
    public static void main( String[] args ) {  
        StdOut.print("Input a integer: ");  
        int i = StdIn.readInt();  
        StdOut.println("The integer you input is:\n"+i);  
    }  
}
```




C:\Windows\System32\cmd.exe

C:\Users\wdx\Desktop\example>javac -cp .;algs4.jar *.java

C:\Users\wdx\Desktop\example>java -cp .;algs4.jar TestStdIn

Input a integer: 123  your input

The integer you input is:

123  program output

Reading from standard input

Reading a lot of integers in one line could be very useful. In the following example, we read an array of integers using:

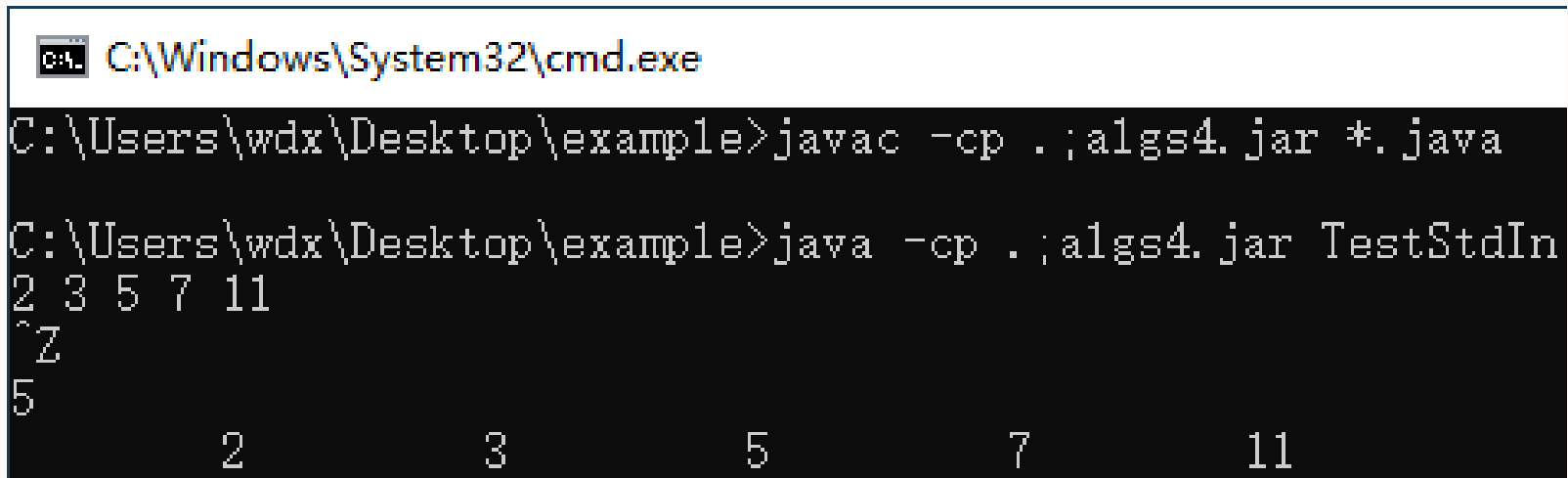
```
int[] arr = StdIn.readAllInts();
```

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

public class TestStdIn {
    public static void main( String[] args ) {
        int[] arr = StdIn.readAllInts();
        StdArrayIO.print(arr);
    }
}
```

Reading from standard input

Example: input 2, 3, 5, 7, 11 in the command prompt and press Ctrl+Z to terminate the input.



```
C:\Windows\System32\cmd.exe
C:\Users\wdx\Desktop\example>javac -cp .;algs4.jar *.java
C:\Users\wdx\Desktop\example>java -cp .;algs4.jar TestStdIn
2 3 5 7 11
^Z
5
2      3      5      7      11
```

Reading from standard input

Maybe you don't want to type a lot of number in the command prompt as input, especially when the array is too long.

You can use a file and redirect the stdin.



The screenshot shows a Windows file explorer window with the following files and their details:

File Name	Modified	Type
algs4.jar	2021/9/3 11:21	JAR 文件
TestStdIn.class	2021/9/8 10:40	CLASS 文件
TestStdIn.java	2021/9/8 10:23	JAVA 文件
input.txt	2021/9/8 10:51	文本文档

Below the file explorer, a Notepad window titled "input.txt - 记事本" is open, showing the following text:

```
文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)
2 3 5 7 11
```

At the bottom, a Command Prompt window (C:\Windows\System32\cmd.exe) shows the execution of the following command:

```
C:\Users\wdx\Desktop\example>java -cp .;algs4.jar TestStdIn < input.txt
```

The output of the command is displayed on the next line:

```
5
    2        3        5        7        11
```

Exercise: read a double from stdin and print to stdout

You have learned the usage of input output apis. Now try to write a program to read a double from stdin and print it to stdout.

You may consider something similar to the following:

```
Input a double: 1.5  
The double value is :  
1.5
```


Reading and writing arrays

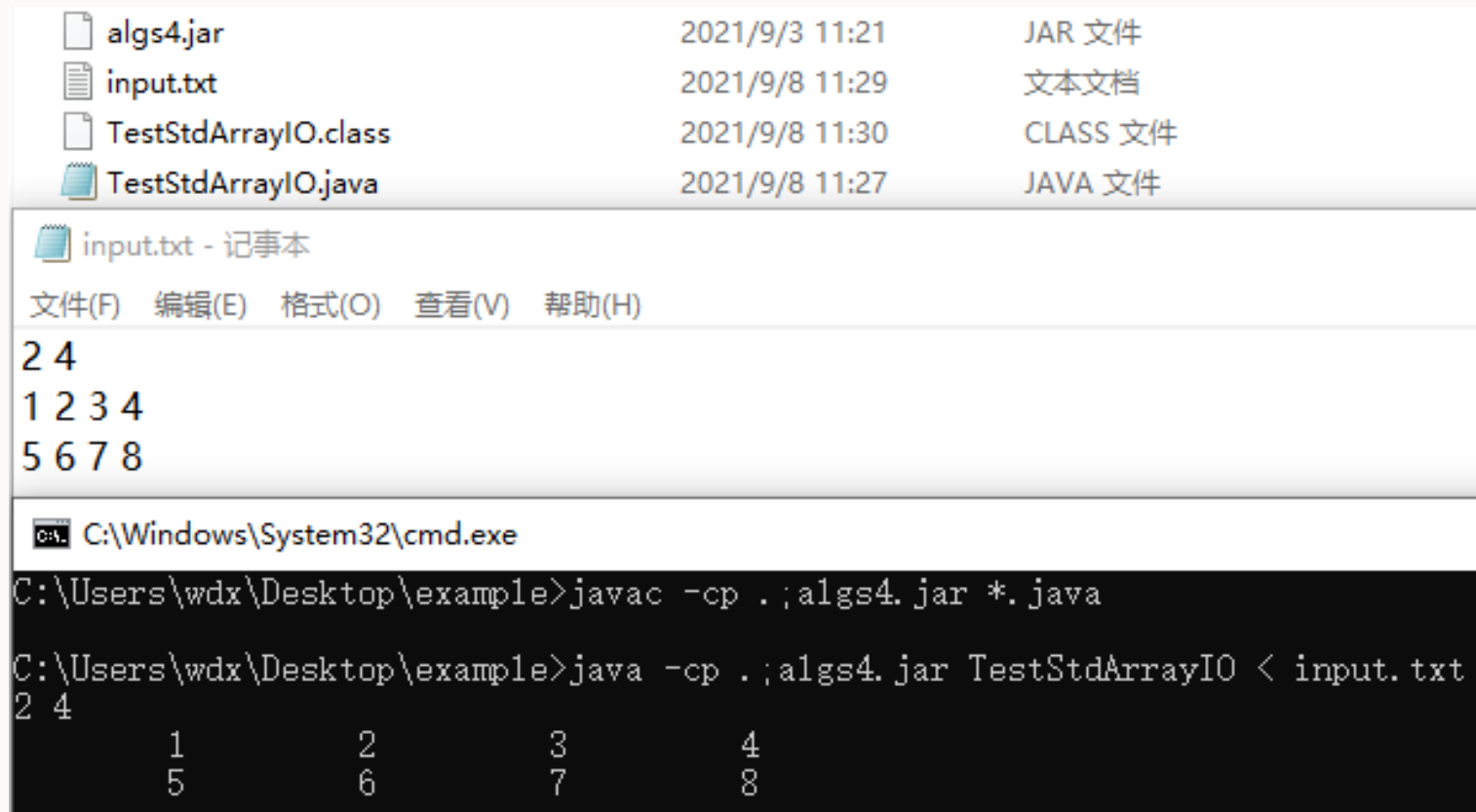
It's simple to read and write 1 dimensional or 2 dimensional array using the StdArrayIO. The follow example reads a 2D array from input and print it to the output.

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

public class TestStdArrayIO {
    public static void main( String[] args ) {
        int[][] arr = StdArrayIO.readInt2D();
        StdArrayIO.print(arr);
    }
}
```

Reading and writing arrays

Example: reads an 2x4 array and print the array.



The screenshot shows a Windows file explorer window with the following files:

File Name	Date/Time	Type
algs4.jar	2021/9/3 11:21	JAR 文件
input.txt	2021/9/8 11:29	文本文档
TestStdArrayIO.class	2021/9/8 11:30	CLASS 文件
TestStdArrayIO.java	2021/9/8 11:27	JAVA 文件

Below the file explorer is a Notepad window titled "input.txt - 记事本" with the following content:

```
2 4
1 2 3 4
5 6 7 8
```

At the bottom is a Command Prompt window titled "C:\Windows\System32\cmd.exe" showing the following commands and output:

```
C:\Users\wdx\Desktop\example>javac -cp .;algs4.jar *.java

C:\Users\wdx\Desktop\example>java -cp .;algs4.jar TestStdArrayIO < input.txt
2 4
      1      2      3      4
      5      6      7      8
```

Draw simple shapes on a windows

The StdDraw class provides simple static methods to draw shapes. It hides all the details of creating JFrame, JPanel, or Graphics2D, so you don't need to remember them.

The majority of the methods in StdDraw consists of two types, methods that draw things on the screen and methods that change the parameters.

Draw simple shapes on a windows

Run the following program and see what happens:

```
import edu.princeton.cs.algs4.StdDraw;  
  
public class TestStdDraw {  
    public static void main(String[] args) {  
        StdDraw.setPenRadius(0.05);  
        StdDraw.setPenColor(StdDraw.RED);  
        StdDraw.point(0.0, 0.0);  
        StdDraw.setPenColor(StdDraw.GREEN);  
        StdDraw.point(1.0, 0.0);  
        StdDraw.setPenColor(StdDraw.BLUE);  
        StdDraw.point(0.0, 1.0);  
  
        StdDraw.line(0.2, 0.2, 0.8, 0.2);  
        StdDraw.rectangle(0.5, 0.5, 0.2, 0.2);  
    }  
}
```

Exercise: determine the coordinate system of the StdDraw

Since most drawing system needs a coordinate system, what could be the coordinate system for StdDraw? How do you know that?

