

CS3354
Object Oriented Programming and Design
Assignment P2
Due: July 20, 2017

July 17, 2017

Submission Instructions:

Please follow carefully the instructions given in P1 with the following additions / modifications

1. Please submit only the source code of your program in Java. In addition, please submit the output file requested. Additionally, please submit an image (in JPEG) that shows the scene obtained through running your program (see screenshot instructions)
2. Follow the style guidelines from the class website.
3. The code should include comments explaining any non-trivial part of the implementation.
4. Use javadoc comments for all of your public elements.

Assignment Spec:

The Collatz Sequence (backward chaining).

Given an integer n the Collatz successors of n are defined to be:

$$B(n) = \begin{cases} n \text{ is even} & n/2 \\ \text{Otherwise} & n * 3 + 1 \end{cases}$$

The trajectory of n , is defined to be the list of successors of n (n is the starting point or SP). Reaching 1 as a successor is considered “success.”

The Collatz Tree (forward chaining).

Consider the following relation:

Given an integer $n > 4$ the Collatz successors of n are defined to be:

$$F(n) = \begin{cases} n \text{ is even AND } ((n - 1) \bmod 3 = 0) & \{2 \times n, (n - 1)/3\} \\ \text{Otherwise} & \{2 \times n\} \end{cases}$$

Consider the following forward chaining procedure:

Given the integer n (goal) and a collection of integers that initially includes only the integer 8.

1. Following a collection management policy (e.g., FIFO), Do:
 - a. Remove an integer (say j) from the collection
 - b. Insert the successor[s] of j to the collection.
 - c. Stop with “success” if n is in the collection

Assignment Instructions:

Your assignment is to write an **interactive Java program** that:

- 1) Obtains the desired chaining mode via a widget (from the user)
- 2) Obtains the number n via a widget
- 3) Obtains a limit (l) on the number of iterations of successors generation via a widget
- 4) Implements the appropriate chaining method; using a sorted list and removing the least element from the list for the forward chaining
 - a. Implement the Comparable<T> and/or Comparator<T> interface(s) in order to sort the list (do NOT write your own sort function, do not use Maps!)
- 5) At each iteration the backward chaining module prints the trajectory followed so far into a new line in a .csv file and the forward chaining module prints the current contents of the collection to a .csv file.
- 6) The program terminates after l iterations of successor generation and reports the status (success or failure) of the run under the l – constraint - via a widget.

Although this is not a good UI practice, for learning purpose, each of these widgets should be a different widget from the java AWT or Swing libraries, e.g., spin-boxes, sliders etc. Each input widget should be connected to an output widget (e.g., an LCD, or text-box) that shows the status of the controlled setup. All the widgets should be in one window that enables “exit” functionality via a push button.

Producing an Image of your scene

The following are instructions for “screen dump”. That is, how to capture the screen / current window into a file under Windows-XP and under Linux.

Print screen under Windows

(From: <http://www.entity.cc/ICONS/print-screen.php>)

To print the contents of the Screen, you must save an image of whatever is displayed on-screen to the clipboard and then paste it into a document where it can be printed.

(Windows Help: Print Screen)

To do this, follow this procedure:

1. Maximize the window you would like to capture.
2. To copy/capture the current window, hold down **ALT + Print Scrn** at the same time. To capture the entire screen, just press **Print Scrn**.
3. Open a new document in MS Word, MS Photo Editor, or Adobe PhotoShop.
4. **Paste** the screen shot by holding down **CTRL** and **V** at the same time.
5. Print the document when you're finished pasting screen shots.

Print screen under Linux

(From: <http://www.sb.fsu.edu/~xray/Manuals/ScreenCapture.html>)

The same location includes more and allegedly better utilities for screen capture)

Using Linux’s native utility ‘xwd/xwud’

Most Linux and UNIX operating systems have native utilities called xwd (x-window dump) and xwud (x-window un-dump) that allow the user to capture either the whole screen or a specific window. In order for the user to capture specific window, issue the following command,

```
‘xwd >myimage.xwd’.
```

Then click with the mouse inside the window of choice (if the whole screen needs to captured, simply click anywhere in the screen). The utility will write out an image in the special ‘xwd’ format. The created image can be viewed either using the related utility ‘xwud’ or using any number of image processing software like, display, gimp, or xv. To view using ‘xwud’ issue the following command,

```
‘xwud -in myimage.xwd<u1:p></u1:’
```

The image can be converted to .jpg or .png using the convert utility.

```
‘convert myimage.xwd myimage.jpg’
```