

# LAB 8 (BINARY TREE, HASHMAP, AND PRIORITY QUEUE)

CSC 172 (Data Structures and Algorithms)

Spring 2018

University of Rochester

**Due Date: Sunday, April 08 @ 11:59 pm**

## Objectives

This lab is designed to give you a chance to play with Binary Trees, HashMap, and Priority Queues. Also, it should aid you to get started with Project 3 if you have not done so yet.

### Problem 1: Level-order printing

Assume the following BTNode class is given.

```
public class BTNode
{
    int data;
    BTNode left;
    BTNode right;
    // Add constructor and/or other methods if required
}
```

Implement a method that prints level-order traversal in a way that nodes of all levels are printed in separate lines. Method signature:

```
void level_order_print(BTNode root);
```

```
      1
     / \
    2   3
   / \   \
  4  5   6
```

Output for above tree should be

```
1
2 3
4 5 6
```

### Problem 2: Reconstructing a Tree

Implement a method that reconstructs a tree from its inorder and preorder traversal (refer: Lecture on Binary Tree Traversal)

Method signature:

```
BTNode reconstruct_tree(int[] inOrder, int[] preOrder);
```

While testing your code, you should perform in-order, pre-order, and level-order traversal on the resultant tree to verify the result.

### Problem 3: Word-count problem

In this problem, you will read each word in a text file and report the number of times each word has appeared in the file. Instead of printing this information on the console, you will write the output to another file.

You must have to use HashMap to solve this problem.

To test your function, you should try the following steps:

First, download the tar files provided for Project 3 and extract the file. You will find a text file, 'alice30.txt' in the extracted folder. Count the number of times each word has appeared in the file (assume, each word is separated by whitespaces. To achieve this, read one line at a time and then split on "\\s+"). You do not need to ignore cases (i.e., consider, Alice and alice as two different strings). A sample entry in the output file may look like:

Alice:221

Method signature:

```
void word_count(String inputFile, String outputFile);
```

### Problem 4: Jesse's cookies

Adapted from: <https://www.hackerrank.com/challenges/jesse-and-cookies/problem>

Jesse loves cookies. He wants the sweetness of all his cookies to be greater than value  $K$ . To do this, Jesse repeatedly mixes two cookies with the least sweetness (sweetness is never negative). He creates a special combined cookie with:

sweetness =  $1 * \text{sweetness of Least sweet cookie} + 2 * \text{sweetness of 2nd least sweet cookie}$ .

He repeats this procedure until all the cookies in his collection have a sweetness  $\geq K$ .

Write a function `Jesse_cookies()` that takes an integer array containing the sweetness of original cookies and a number  $K$  representing the minimum required sweetness. Return the number of operations (mixing) required to give the cookies a desired sweetness. Return -1 if this is not possible.

Method signature:

```
int Jesse_cookies(int[] cookies, int k);
```

### Submission

Save the four files as `Lab8P1.java`, `Lab8P2.java`, `Lab8P3.java`, and `Lab8P4.java`. Hand in these four files and a README file at the appropriate location on the Blackboard system at `learn.rochester.edu`. You should hand in a single zip (compressed archive) `Lab8zip`. The README file should state your and your team member's name and any other pertinent information.

### Grading (100 pts)

Each problem is worth 25 points.

We will keep on updating this section to address your questions. Please check the lab description periodically.

- You must not submit any other files except the four stated.
- You must adhere to the naming of your classes and methods.
- Each file must contain a `main()` method that calls the respective method to test its functionality. We may use cases to test your files though.
- For Problem 3, you can use `[^a-zA-Z]` for splitting, but it's not a requirement. It's fine if you keep special characters as it is, i.e., Alice, Alice's, Alices, are three different words.
- Once again, TAs may or may not use the same main method that you wrote.