Lab 03: Efficiently Enumerating Subsets, Part 2
COSC 290 - Fall ’22

<table>
<thead>
<tr>
<th>Starter File(s)</th>
<th>L03_starter.zip (2 .java files, 1 .txt file)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Submission</td>
<td>Upload only the following file(s) to Moodle:</td>
</tr>
<tr>
<td></td>
<td>• EnumeratingSubsets.java</td>
</tr>
<tr>
<td></td>
<td>• Lab03Tester.java</td>
</tr>
<tr>
<td></td>
<td>• lab03_analysis.txt</td>
</tr>
<tr>
<td>Due Date</td>
<td>Mon, October 3rd at 10PM for all lab sections</td>
</tr>
</tbody>
</table>

1 Overview

This lab is a continuation of the concepts you encountered last week, namely working with the Java implementation of Set along with a Java abstraction of subsets, as well as iterating over and manipulating Sets. The final part of the lab will see you utilizing your implemented functionality to analyze the growth of powersets of various cardinalities.

2 Pre-Lab Questions

At the end of last week’s lab (Efficiently Enumerating Subsets, Part 1), you were asked to come up with responses to the following questions. If you have not done so, complete them before the start of this week’s lab:

1. Sets can be defined via enumeration or abstraction.
   Let’s define S by enumeration:  
   \[ S := \{1, 2, 3\} \]
   and let’s define T be abstraction:  
   \[ T := \{(x + 1) \mod 4 : x \in S\} \]
   What are the elements of T?

2. Suppose S := ∅. What is \(\mathcal{P}(S)\)?

3. Consider A := \{a, b, c\}
   What is \(\mathcal{P}(A - \{a\})\)? See if you can observe a relationship between \(\mathcal{P}(A)\) and \(\mathcal{P}(A - \{a\})\)

4. Given a set S of arbitrary size and let x be some element in S.
   Let \(P_x := \mathcal{P}(S)\)
   Define \(T := S - \{x\}\)
   and define \(P_t := \mathcal{P}(T)\)
   Express \(P_x\) in terms of \(P_t\) and x using set operations (union, intersection, difference) and set abstraction (see first question).
   Your answer to the last question provides a hint on how you might implement power set generation using recursion (think of \(P_t\) as the result of a recursive call on a smaller input).

3 Your Task

Your task for this lab is split into two parts – first you will implement functionalities related to the generation and analysis of powersets, then you will use this functionality to run an experiment and record your observations.
3.1 Part 1: Powerset Generation and Analysis

Begin by implementing the following two methods in the provided `EnumeratingSubsets.java`:

- **generatePowerset**: takes a Set of Strings `s` and returns the powerset of `s` (which will be a Set of Set of Strings). This function should not modify its argument Set and **must be implemented recursively** – some initial code is provided to help you get started!
  
  If you are having trouble thinking about designing this function recursively, review the pre-lab questions, and also the Lab 00 code / writeup (particularly Section 2: Thinking Recursively)! If a null argument is passed, this function should throw an `IllegalArgumentException`.

- **countCardinalities**: takes a Set of Sets of Strings `s` and returns an array of ints indicating the count of subsets of each particular cardinality in `s`. More specifically, in the returned array, the value of each index in the array represents the number of subsets in `s` that have a cardinality of that respective index.

  For example, if `s` represented the following Set of Sets:
  
  ```
  {{"b","c"}, {"a"}, {"d","e"}, {"f","g","h","i"}}
  ```

  then `countCardinalities(s)` would return an array containing:
  
  ```
  {0, 1, 2, 0, 1}
  ```

  as `s` contains 1 subset of cardinality 1, 2 subsets of cardinality 2, and 1 subset of cardinality 4. Given an empty set of sets, this function should return an array with a single zero (i.e.: `{0}`). If a null argument is passed, this function should throw an `IllegalArgumentException`.

3.2 Part 2: Experiment, Observe, and Analyze

Once you have implemented and adequately tested the two functions above, complete the following:

- In the main of your `Lab03Tester` implement code that outputs the result of `countCardinalities` given the powerset of sets with cardinalities 1 through 10. More specifically:
  
  - make ten calls to `countCardinalities`, the first call passing the powerset of a Set of cardinality 1. *(Note: don’t hard-code ten calls to `countCardinalities`, your code should be more elegant than that!)*
  - print out the result of each call to `countCardinalities` on its own line
  - your output should only contain numeric values separated by commas (no brackets or curly braces).

  For example, the fifth line of your output (powerset of cardinality 5 Set) should look like the following:
  
  ```
  1, 5, 10, 10, 5, 1
  ```

- Next copy and paste all ten lines of your output into the designated area in the provided `lab03_analysis.txt`.

- Now, you will create a chart to visualize the distribution of these powersets and their cardinalities. To do so:
  
  - Navigate your web browser to `drive.google.com`. Log into your `@colgate.edu` Google account (if you’re not logged in already).
  - Click the + New button in the upper left corner, then select *Google Sheets* -> *Blank Spreadsheet*.
– Once again copy all ten lines of your experiment output from your lab03_analysis.txt. Right-click on cell A1 and select Paste special -> paste values only. (Note: Google Sheets may do some auto-formatting with the first two rows, thinking they are calendar dates. If so, don’t worry, you’ll fix this in a couple steps).

– Next, highlight all ten rows of your data. From the top menu bar, select Data -> Split text to columns. This will split each row’s values into their own column, as shown in the pictures below (Note: the numbers in the pictures below are fake, sorry!):

Before:

After:

– Double check the values in your first two rows – if these were auto-formatted earlier, manually correct the values now.
Now, you can visualize your data into various graphs by highlighting all the cells containing your data, and selecting Insert -> Chart. Experiment with different chart types and the other options available here.

When you are ready, in the designated area of your lab03_analysis.txt, write one to two short paragraphs of your observations. What patterns do you see? Does any of this data surprise you? What becomes more apparent to you as a result of the chart visualization?

4 Submission

See the top of this document for your lab section’s due date/time. Don’t forget, you will be evaluated on the quality of the test cases provided in your Lab03Tester.java.

When uploading your submission to Moodle submit only the files listed below – please do not upload a .zip or any .class/.java~ files:

- EnumeratingSubsets.java
- Lab03Tester.java
- lab03_analysis.txt