CSC172 LAB 5

COMBINATORICS 2

1 Introduction

The labs in CSC172 will follow a pair programming paradigm. Every student is encouraged (but not strictly required) to have a lab partner. Labs will typically have an even number of components. The two partners in a pair programming environment take turns at the keyboard. This paradigm facilitates code improvement through collaborative efforts, and exercises the programmers cognitive ability to understand and discuss concepts fundamental to computer programming. The use of pair programming is optional in CSC172. It is not a requirement. You can learn more about the pair programming paradigm, its history, methods, practical benefits, philosophical underpinnings, and scientific validation at http://en.wikipedia.org/wiki/Pair_programming

Every student must hand in their own work, but every student must list the name of their lab partner if any on all labs.

This lab has six parts. You and your partner(s) should switch off typing each part, as explained by your lab TA. As one person types the lab, the other should be watching over the code and offering suggestions. Each part should be in addition to the previous parts, so do not erase any previous work when you switch.

The textbook should present examples of the code necessary to complete this lab. However, collaboration is allowed. You and your lab partner may discuss the lab with other pairs in the lab. It is acceptable to write code on the white board for the benefit of other lab pairs, but you are not allowed to electronically copy and/or transfer files between groups.

Look up the Java documentation on the BigInteger class for this assignment.
http://docs.oracle.com/javase/7/docs/api/java/math/BigInteger.html

2 Combinatorics

The goal of this lab it to gain familiarity with the practical aspects of combinatorics.

1. **ORDERING WITH IDENTICAL ITEMS** : Sometimes we wish to select only some of the items in a set where some of the items are indistinguishable but the order does matter. If there are n items divided into k groups of sizes $i_1, i_2, i_3, \ldots i_k$ where items in the same group are not

distinguishable, then number of different distinguishable orders of the ***n*** objects is

$$\frac{n!}{\prod_{j=1}^{k} i_j!}$$

Write a Java method to recursively compute this value and use it to answer the following questions (in the main method).

1. Count the number of anagrams of the following words (a) **error** (b) **street** (c) **allele** (d) **Mississippi** .

2. **DISTRIBUTING INDISTINGUISHABLE OBJECTS INTO BINS** : Sometimes we wish to distribute indistinguishable objects into distinguishable categories (bins). The number of ways to distribute ***n*** items distributed into ***m*** bins is

$$\binom{n+m-1}{n}$$

Write a Java method to recursively compute this value and use it to answer the following questions (in the main method).

1. How many ways can we distribute (a) **six apples to four children** (b) **four apples to six children** .

3. **DISTRIBUTING DISTINGUISHABLE OBJECTS INTO BINS** : Sometimes we wish to distribute distinguishable objects into distinguishable categories (bins). We can extend the previous formula to allow for distribution of into ***m*** bins of a collection of ***n*** objects that fall into ***k*** different classes. The number of ways to distribute ***n*** items that fall into ***k*** classes into ***m*** bins is

$$\frac{(n+m-1)!}{(m-1)!\prod_{j=1}^{k} i_j!}$$

Write a Java method to compute this value and use it to answer the following questions (in the main method).

1. How many ways can we distribute (a) **six apples and three pears to five children** (b) **two apples, five pears and six bananas to three children** .

3 Hand In

Hand in the source code from this lab at the appropriate location on the BlackBoard system at my.rochester.edu. You should hand in a single zip file (compressed archive) containing your source code, README, and OUTPUT files, as described below.

1. A plain text file named README that includes your contact information, your partner's name, a brief explanation of the lab (A one paragraph synopsis. Include information identifying what class and lab number your files represent.), and one sentence explaining the contents of all the other files you hand in.
2. Several Java source code files representing the work accomplished for this lab. All source code files should contain author and partner identification in the comments at the top of the file
3. A plain text file named OUTPUT that includes author information at the beginning and shows the compile and run steps of your code. The best way to generate this file is to cut and paste from the command line.