Quantify the theoretical complexity of your algorithm for finding subsequence matches and Regions, and validate the complexity based on empirical measurements. Analyze the time complexity of WeightedList.

**Project Analysis 1: Gene Sequencing**

Now that you have completed the first project, write a short (3-5 pages) paper analyzing your design and implementation. Specifically, you should answer the following questions, each in a different section of the paper:

1. What is your algorithm for finding the longest common subsequences in a pair of strings? For simplicity, you may ignore the problem of insufficient memory to store the strings; that is, you do not need to include details about breaking the large strings into fixed-sized chunks.  
     
   In your narrative, you should include enough pseudo-code or Java code to summarize your algorithm.
2. What are the advantages and disadvantages of WeightedList when compared to ArrayList and LinkedList? Please summarize your algorithm for adding an item to the WeightedList and analyze its time complexity. When might this data structure be preferred over the others?
3. What is your algorithm for finding the next Region within a given genome snippet? Please summarize your algorithm and analyze its time complexity. Include enough code to illustrate your analysis.
4. Use the Runner class to time any of the algorithms you have implemented for inputs of various sizes. For instance, you might run the longestCommonSubsequence method on snippets of size 100, 10,000, 1,000,000, and so on. You can use System.currentTimeMillis to time the execution as shown below. Do these timing measurements match the complexity calculation you performed? Why or why not?

**long** start = System.currentTimeMillis();

doSomeStuffHere();

**long** end = System.currentTimeMillis();

System.out.println("That took " + (end - long) + " ms");

Complete this paper according to the following requirements:

* You may work with **one** other person on this paper. You should look at each others' code that was submitted and pick one person's implementation to analyze. You don't have to use the same person's code for every section. E.g., you could use person A's code for parts 1 and 4, and the other person's code for parts 2 and 3.
* Use the [**p1-analysis.tex**](https://w3.cs.jmu.edu/kirkpams/240/f19/projects/p1/p1-analysis.tex) LaTeX template. It compiles into [**p1-analysis.pdf**](https://w3.cs.jmu.edu/kirkpams/240/f19/projects/p1/p1-analysis.pdf). Note that the LaTeX source has a lot of samples built into it that illustrate how to format various parts of your paper using LaTeX. You can load this file into [**Overleaf**](https://www.overleaf.com/), a browser-based application for editing LaTeX documents.
* Note that this is a paper. As such, you should take care to write clearly in sentences, proofread, fix typos, etc.
* Please print a copy and bring it to class on Wednesday, September 25. You only need to bring one copy per pair.