*1. Whether any of your classes have known bugs or other problems that we should*

*know about.*

All my classes have the correct functionality outlined in the spec.

*2. Whether or not each method satisfies our big-O requirements, and if not, what*

*you did instead and what the big-O is for your version.*

All of my methods satisfy the big-O requirements.

*3. How two of your methods work — use high-level pseuocode to describe them:*

*• Trie's find() method*

Trie’s find class my recursive find function.

* Check if I am done with the input key or am at the bottom of the trie (base cases)
* If the first letter of key is equal to the current node
  + If there is a value here, add it to output
  + Recursively call find on this node’s children
* If the first letter of key is not equal to the current node, but exact match false
  + Set exact match to true
  + If there is a value here, add it to output
  + Recursively call find on this node’s children

*• GenomeMatcher's findGenomesWithThisDNA() method*

* Extract min search length letters from the fragment
* Call Trie’s find on that fragment
  + Loop through the hits
    - Search the unordered map for the appropriate genome for this hit
      * Extract the string to represent the longest possible string from hit (fragment length)
      * Loop over the maximum size fragment and compare character by character
        + If exact match is false and mistake is made, set exact match to true
        + If there is a mistake, stop looking
      * Store the longest hit and its position in unordered map
* Loop through the genomes
  + If we have saved a length and position for the genome, add it to the vector
* If the vector is empty, return false, otherwise true