CptS233 Programming Assignment #2 -
String Duplicate Detection with Hashtables (via HashMap)

For this assignment, you will be tasked with using a hash table to detect duplicate strings in a vector of strings. The code already has the ArrayList<String> data and the function call for your hash table-based algorithm implementation to go into. You DO NOT need to implement a hash table object of your own. You should be using the Java Library's HashMap<> as your hash table, though if you want to write your own, go nuts.

There is another duplicate detection algorithms already in the code for your reference and comparison.

(1) Brute force pair comparisons: a true brute force kind of algorithm that compares every element in the vector to every one, which is a O(N^2) kind of algorithm. More precise bound would be about (N^2/2 - N/2).

Your code should use the Java Library's hash table, HashMap. The documentation for it may be found here:

https://docs.oracle.com/javase/8/docs/api/java/util/HashMap.html

The basic concept is that you take each string in the input vector and:

Check to see if it's already in your hash table.
    If it is, check to see if it's in your duplicates vector already
        If it's not already in your duplicates vector, add it
    otherwise
        Ignore it - **our vector of duplicates can't have duplicates itself**
Add it to the hash table - double adds won't be a problem

Now, the HashMap takes two types: HashMap<Key, Value>. In our situation we only have a key that we care about, which is a string. Since we need some kind of Value you can either use the string in both positions, or just put something else small there. A boolean would make sense, but I just used HashMap<String, String> and put the string that I was searching for duplicates into both the Key and Value fields, but you can do what you like with the values.

The most interesting interfaces you'll run into by reading the library documentation (see URL above) are likely:

- containsKey(Object key); -- Returns boolean if the map already contains the key given
- put(K key, V value); -- Adds a key value pair to the hash table

The code has been pushed to a new branch in your git repository and a merge request created to merge PA2-StringDups branch into master. You should do that ASAP! If you work on master before merging this branch in you could end up with merge conflicts. These would need resolving. They're unlikely in this

case, but worth avoiding if possible. There will also be a separate merge request for the CI file to enable the tests.

Grading

Your submission will be graded based on the following:

1. [90] Your modifications cause no runtime issues, your HashMap based hash table properly finds duplicates and the proper results as shown in the tests when you run make test.
2. [10] Your modifications contain good style. For example,
    - You provide meaningful variable names
    - You provide sufficient and meaningful comments
    - Your code is well structured

Submission Process

Commit and push your code to your git repo. Then put a small file on Blackboard for this assignment to let the TAs know you're done and ready to be graded. The small file truly can be anything. I hear the TAs like funny cat pictures.

CI Update Note:

I shall be pushing the CI updates separately for this assignment.