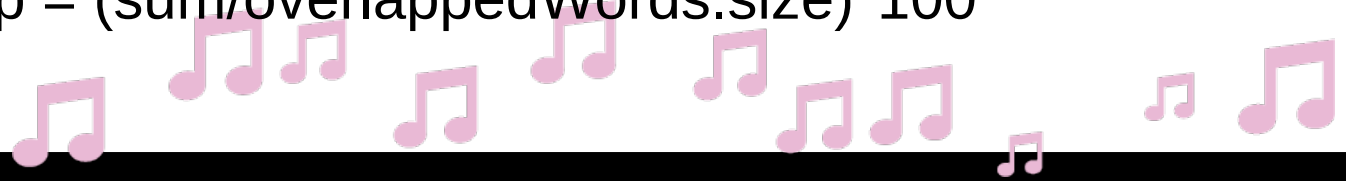
Comprehensive Programming Practice 1

Tasks:

- Read two text files
- Find overlap words between the two files
- Find unique words between those two files
- Calculating percentage of the overlaps

Procedures:

- Create two ArrayList: list1, list2
- Read two text files using scanner
- Put the strings of the two files into the two ArrayLists consecutively
- Sort(list1), sort(list2)
- getRidOfDuplicates(list1), getRidOfDuplicates(list2)
- Sum = list1.size() + list2.size()
- Merge both lists into list1: list1 = merge(list1, list2)
- sort(list1)
- ArrayList uniqueWords = getUniqueWords(list1)
- ArrayList overlappedWords = getOverlappedWords(list1)
- Percentage_of_overlap = (sum/overlappedWords.size)*100



getUniqueWords(ArrayList list)

- (We assume the list is sorted.)
- uniqueWords = new empty list
- current = list.get(0)
- For i = 1 to list.size() :
  - If list.get(i) is not equal to current :
    - current = list.get(i)
    - If i<list.size()-1 and current is not equal to list.get(i+1) :
      - UniqueWords.add(current)
    - Else if i==list.size()-1 :
      - UniqueWords.add(current)
- Return uniqueWords

Source code:getUniqueWords(ArrayList list)

```
/**
 * Method to get all the unique words
 * @param list
 * @return uniqueWords
 */
static ArrayList<String> getUniqueWords(ArrayList<String> list) {
    ArrayList<String> uniqueWords = new ArrayList<String>();
    String curr = list.get(0);
    for(int i = 1, len = list.size(); i < len; i++) {

        if (!list.get(i).equals(curr)){
            curr = list.get(i);
            if(i < len - 1 && !curr.equals(list.get(i+1)) ){
                uniqueWords.add(curr);
            }
            else if (i == len-1) {
                uniqueWords.add(curr);
            }
        }
    }
    return uniqueWords;
}
/**
```

getOverLappedWOrds(ArrayList list)

- overlaps = new empty list
- Prev = list.get(0)
- For i=1 to list.size() :
  - current = list.get(i)
  - If current equals prev:
    - overlaps.add(prev)
  - Prev = current
  - Else if i==list.size()-1
    - Overlaps.add(prev)
- Return overlaps

Source code:

```java
/**
 *
 * @param list
 * @return
 */
static ArrayList<String> getOverlapWords(ArrayList<String> list) {
    ArrayList<String> overlaps = new ArrayList<String>();
    String prev =list.get(0);
    for(int i = 1; i < list.size(); i++) {
        String curr = list.get(i);
        if (curr.equals(prev))
            overlaps.add(prev);
        prev = curr;
        if (i == list.size()-1)
            overlaps.add(prev);
    }
    return overlaps;
}
```

Time Complexity:

- Methods:
  - fileToArray(Scanner, ArrayList)
  - Collections.sort(list)
  - getRidOfDuplicates(ArrayList)
  - mergeTwoArrayList(list1, list2)
  - getUniqueWords(ArrayList)
  - getOverlappedWords(ArrayList)

Time Complexity:

- Time complexity of each method
- No method with time complexity more than linear time
- Thus by rule 2: $T_1(n)+T_2(n)=\max O(f(n)),O(g(n))$
- The time complexity is Linear.

THANKS EVERYONE FOR LISTENING!!