Part1：

CustomLinkedList.java | Node.java | little_women.txt | DuplicateWords.java | Urll

```java
package sets;
import java.util.*;
import java.io.*;

public class DuplicateWords {

    public static String cleanString(String originalString) {
        String cleanedString = originalString.replaceAll("\\p{Punct}", "");
        cleanedString = cleanedString.replaceAll("[“’”]+", "");
        return cleanedString.toLowerCase();
    }

    public static void main(String[] args) throws IOException {
        String path = args.length > 0 ? args[0] : "little_women.txt";

        Set<String> once = new HashSet<>();
        Set<String> dup = new HashSet<>();

        try (BufferedReader br = new BufferedReader(new FileReader(path))) {
            String line;
            while ((line = br.readLine()) != null) {
                String[] words = line.split("\\s+");
                for (String w : words) {
                    w = cleanString(w);
                    if (w.isEmpty()) continue;

                    if (dup.contains(w)) continue;
                    if (once.contains(w)) {
                        once.remove(w);
                        dup.add(w);
                    } else {
                        once.add(w);
                    }
                }
            }
        }

        System.out.println("number of duplicate words: " + dup.size());
        System.out.println("number of single words: " + once.size());
    }
}
```

Problems | Javadoc | Declaration | Coverage | Console | Eclipse IDE for Java Develope

<terminated> DuplicateWords [Java Application] D:\eclipse-java-2025-06-R-win32-x86_64\eclipse\plugins\org

```
number of duplicate words: 253
number of single words: 382
```

Part2：

CustomLinkedList.java | Node.java | little_women.txt | UrlLog.java | DuplicateWords.java

```
                line = line.trim();
                if (line.isEmpty()) continue;

                // skip header line if present
                if (first) {
                    first = false;
                    if (line.toLowerCase(Locale.ROOT).startsWith("timestamp,")) {
                        continue;
                    }
                }

                // split CSV: timestamp,userid,url
                String[] parts = line.split(",", 3);
                if (parts.length < 3) continue;
                String timestamp = parts[0].trim(); // not used but kept
                String user = parts[1].trim();
                String url = parts[2].trim();

                // record in userToRecent
                ArrayDeque<String> dq = userToRecent.computeIfAbsent(user, k -> new ArrayDeque<>());
                if (dq.size() == MAX_RECENT) dq.pollFirst(); // remove oldest
                dq.addLast(url);

                // count url frequency
                urlCount.merge(url, 1, Integer::sum);
            }
        } catch (IOException e) {
            System.err.println("Cannot read file: " + path);
            e.printStackTrace();
            return;
        }

        // --- print user recent logs ---
        System.out.println("Recent URLs per user:");
        for (Map.Entry<String, ArrayDeque<String>> e : userToRecent.entrySet()) {
            List<String> list = new ArrayList<>(e.getValue());
            System.out.printf("%s -> %s%n", e.getKey(), list);
        }

        // --- print URL visit counts ---
        System.out.println("\nURL Visit Counts:");
        List<Map.Entry<String, Integer>> sorted = urlCount.entrySet()
                .stream()
                .sorted(Comparator.comparing(Map.Entry<String, Integer>::getValue, Comparator.reverseO
                        .thenComparing(Map.Entry::getKey))
                .collect(Collectors.toList());

        int maxLen = sorted.stream().mapToInt(a -> a.getKey().length()).max().orElse(0);
        for (Map.Entry<String, Integer> e : sorted) {
            System.out.printf("  %-" + maxLen + "s : %d%n", e.getKey(), e.getValue());
        }
    }
}
```

Problems | Javadoc | Declaration | Coverage | Console | Eclipse IDE for Java Developers 2025-09 Release

<terminated> UrlLog [Java Application] D:\eclipse-java-2025-06-R-win32-x86_64\eclipse\plugins\org.eclipse.justj.openjdk.hotspot

```
Recent URLs per user:
u001 -> [/cart, /orders, /logout, /login, /checkout]
u002 -> [/home, /products, /checkout, /orders, /logout]
u003 -> [/checkout, /logout, /login, /home, /products]
u004 -> [/products, /cart, /orders, /logout, /checkout]

URL Visit Counts:
  /products : 8
  /checkout : 7
  /home     : 7
  /logout   : 7
  /login    : 5
  /orders   : 5
  /cart     : 4
```

Part3：

CustomLinkedList.java | Node.java | little_women.txt | UrlLog.java | DuplicateWor

```
166         for (Object e: c) if (!contains(e)) return false;
167         return true;
168     }
169
170     @Override
171     public boolean removeAll(Collection<?> c){
172         boolean ch = false;
173         for (Object e: c){
174             boolean r;
175             do { r = remove(e); ch |= r; } while (r);
176         }
177         return ch;
178     }
179
180     @Override
181     public boolean retainAll(Collection<?> c){
182         boolean ch = false;
183         Entry<T> dummy = new Entry<>(null);
184         dummy.next = head;
185         Entry<T> p = dummy, cur = head;
186         while (cur != null){
187             if (!c.contains(cur.v)){
188                 p.next = cur.next;
189                 if (cur == tail) tail = p;
190                 size--; ch = true;
191             } else {
192                 p = cur;
193             }
194             cur = cur.next;
195         }
196         head = dummy.next;
197         if (head == null) tail = null;
198         return ch;
199     }
200
201     // ---- simple benchmark ----
202     public static void main(String[] args){
203         final int COUNT = 1_000_00; // 100k (1e5) to keep it fast
204         ThreadLocalRandom r = ThreadLocalRandom.current();
205
206         long t0 = System.nanoTime();
207         Queue<Integer> lib = new LinkedList<>();
208         for (int i=0;i<COUNT;i++) lib.add(r.nextInt());
209         long t1 = System.nanoTime();
210
211         long t2 = System.nanoTime();
212         Queue<Integer> cus = new CustomLinkedList<>();
213         for (int i=0;i<COUNT;i++) cus.add(r.nextInt());
214         long t3 = System.nanoTime();
215
216         System.out.println("time for library LL: " + (t1 - t0)/1_000_000.0 + " ms");
217         System.out.println("time for custom  LL: " + (t3 - t2)/1_000_000.0 + " ms");
218     }
219 }
220
```

Problems | Javadoc | Declaration | Coverage | Console | Eclipse IDE for Java Developer

```
<terminated> CustomLinkedList [Java Application] D:\eclipse-java-2025-06-R-win32-x86_64\eclipse\plugins\or
time for library LL: 3.2846 ms
time for custom  LL: 2.3417 ms
```