

Project 2

Preliminary Drawings

CSc 136

Purpose: Prepare for Project 2 by creating drawings showing the method to be used for ordering the list and tabulating duplicate tokens.

Submit: The D2L *Project 2 Drawings* dropbox.

Description: In Project 2, you will use a specialized version of the selection sort for ordering a list of WordRec objects and tabulating duplicates. New tokens will be placed at the end of the used elements of the array until it becomes full or the file is exhausted. At that time the array will be sorted, and during the sort, when a duplicate is about to be placed, its match (which will be right before where the duplicate is about to be placed; do you understand why this is? You will see it clearly when you do the drawing) will have its counter incremented. The array will be adjusted to place the duplicate element on the garbage (unused elements) side of the *numElts* counter, and *numElts* will be updated to reflect the one less active element in the array.

This procedure will be repeated until the file's contents are exhausted, i.e. once the array has been sorted and duplicates processed, there may again be space in the array, so the process of filling the array and processing the data can repeat. This continues until the file is exhausted or the array remains full after the sort/remove duplicates operation. If the array is full following the process of sorting and processing duplicates, tokens will be read solely for the purpose of identifying and recording duplicate tokens. All non-matching tokens are simply not placed into the array, i.e. discarded.

Create two drawings to demonstrate removal of duplicates. Work with an array of WordRec with a five element capacity and a file of strings that has at least 10 elements, six unique, where at least two are duplicates, and at least one duplicate appears after the sixth unique element is discarded (for lack of room).

1. Given that the array, with capacity 5, can be partially filled, show the process that will be carried out during the first sort/remove duplicates operation when a duplicate is found and needs to have the matching element(already sorted in position)'s counter incremented. Handle all aspects, showing what changes, where things go, etc;

This drawing shows this occurrence at a time when the array has just become full for the first time, and is not yet sorted, with at least one duplicate element stored, and the element matched must not be in the first position.

Show first the data file and the full array after reading the first five tokens from the file. Then, show how the first element is placed where it belongs. During placement of subsequent elements, detail the first encounter with a duplicate element, where it causes its match to have its counter incremented and the duplicate element is effectively removed from the active list, with its element counter updated.

2. For the same data file, show the array after each iteration of the sort/remove duplicates process at the point in time after the array became full and sort/duplicate removal completes. In short, the drawing shows the progression from empty until the file is exhausted. The final drawing shows the array, not necessarily at capacity (there may not be enough unique tokens to fill it), with all elements unique and multiplicities properly recorded.

Notes:

- You are to use a data file of your own creation. No sharing data files.
- There are special cases. The first person in each section to identify one of the special cases, which are related, by sending me an email, will receive 3 bonus points on the project.
- You will receive timely feedback once you submit your drawings. No project will be graded without first having drawings approved.
 - **You may not start coding until your drawings are approved.**

Deliverable

- Drawings, in the *Project 2 Drawings* dropbox in D2L