lwords

First, read `list.h` to understand the API. Focus on the examples given in the big docstring in the beginning of the file.

Next, thoroughly read through the data structures and methods in `word_count.h`. In particular, pay attention to the `word_count_t` and `word_count_list_t` structs. You may find it beneficial to see the compiler flags used in the `Makefile` and how that affects the struct definitions.

Finally, complete `word_count_l.c` to properly implement the word count API given in `word_count.h`. **You must use the Pintos list API**. After you finish making this change, `lwords` should work properly (i.e. exhibit the same behavior as frequency mode of words).

The `wordcount_sort` function sorts the wordcount list according to the comparator passed as an argument. Although `lwords` uses the less count function from `word_helpers.h` as the less argument, the wordcount sort function should be generic enough to work with any valid comparator passed in as the less argument. For example, passing the `less word` function from `word_helpers.h` as the `less` parameter should. Check out some basics on function pointers if you're having trouble understanding and writing the syntax.
