Burt Rosenberg

Problem Set 5

OUT: 23 SEPTEMBER, 1993
DUE: 1 OCTOBER, 1993

Goals

Perfection in understanding and manipulating linked lists.

Reading Assignment

Read Chapter 4 from *Algorithms.* This pertains to the class discussion on Trees.

Programming Assignment

Last week's program is too slow! The problem is that the length of a search depends linearly on the depth of the found item, meanwhile, under the insert-at-head scheme, common words such as "a" and "the" tend to be at the end of the list! Let's remedy this incongruity.

There are two approaches. Both are heuristics. A *heuristic* is an algorithm which tends to work although cannot be guarantied to work. Your assignment is to implement the *move-to-front* heuristic. For extra credit, also implement the *insert-at-tail* heuristic, test, compare and comment on relative performance.

Move-to-front acts the same as insert-at-head for words not found in the list — they are added to the head of the list. However, if a word is on the list, move-to-front increments its count *and* moves the word to the head of the list. Common words thus always cluster around the list's head.

◇ ◇ ◇

EXTRA CREDIT: In insert-at-tail, new words are inserted at the end of the list, rather than its head. The tendency will now be for unusual words to find themselves at the back of the list. However, a diabolically constructed file, such as a long text file with the words sorted, will make insert-at-tail run *worse* than insert-at-head. But OK, that's why it is called a *heuristic*.

The insert-at-tail implementation will require some rethinking the list data-type. We are all interested to see what you come up with!