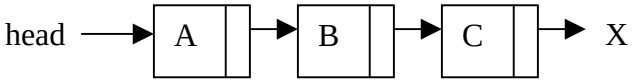
Homework 1
69 points possible

This homework will be turned in electronically. For the homeworks only you will need to turn in either a postscript file or a pdf (but this does not mean that you can't still use Word for creating a postscript file). It would be wonderful if you did not do the homework in Word but rather learned LaTex or DIA, however, Word is just fine as well. Go to http://www.cs.ucr.edu/cs14/cs14_04win/tutorials/postscript.html for information on creating postscript and pdf files. DO NOT wait until the last minute to create your ps/pdf because late homeworks will not be accepted no matter what. Please make sure that you view/print your ps/pdf BEFORE you turn your homework in. Corrupted ps/pdf files WILL NOT be allowed to be resubmitted.

For 10% extra credit, form a study group of at least 3 people. Be sure to include the names of your partners and how long you worked on the homework together. Remember that you must work with at least 2 other people to form a group of three and you must work for at least 2 hours together (minimum of 1 hour sessions). See the main course web page for restrictions on homework collaboration.

1. (12 pts) Draw a singly linked list that contains three items A, B, and C. The head pointer points to the A item. Now explicitly write out the pointer operations required for each of the following parts. You should have no loops, only a set of statements for explicit pointer operations. (Each part operates on the original list):



a) insert D at end
b) insert D before B
c) insert D at head
d) delete B
e) delete C

2. (7 pts) Write a recursive function that returns the number of occurrences of some specific item in a linked list. State any assumptions you make. Give the Big-Oh notation for the function.

3. (10 pts) A self-adjusting list is like a regular list, except that all insertions are performed at the head, and when an element in accessed through Search, it is moved to the head of the list without changing the relative order of the other items. The Search function returns either TRUE or FALSE based on whether the item was found. Write the Insert and Search functions for a self-adjusting list implemented with an array. The search function should find and move all occurrences of the item. (Your algorithm should not take into consideration the type of item in the list.)

4. (10 pts) Write the Insert and Search functions for the self-adjusting list implemented with a singly linked list.

5. (5 pts) Program A has a running time of $n^2 - 1500$, where n is the size of the input. Program B has a running time $400n + 16$. For what values of n does program A run faster? For what values of n does program B run faster? Show your work

6. (10 pts) Give the Big-Oh notation for each of the following functions:

a) n^2+5n
b) $3n^2+6n$
c) $(n+7)(n-2)$
d) $100n + 12$
e) $7n+2n^2$

7. (15 pts) Write a program that takes a single command line argument *m* (see the tutorials for information regarding command line arguments). The program should initialize a double variable *x* to 1, and then compute the following function 2000^m times: x = x + (26.98 + 1)/26.98 – 1. Use the *pow* function in the math library to compute 2000^m. (NOTE, you will need to include cmath and you will need to provide the command line option –lm to g++ to link the math library) Run the program for *m* equal to 1 and 2 and measure the runtimes using the Unix "time" command (time executable name arg1). Run your program on a local PC in lab. Based on these runtimes, **estimate** the runtime for *m* = 3 (**do not** run your program for *m* = 3). This example demonstrates the tremendous differences between algorithms of complexity n, n^2, and n^3. Paste a copy of your source code into your homework. Include your measured runtimes for *m* equal to 1 and 2 and your computation and final estimate of a runtime for *m* = 3.