

CSE 326  
Winter 2008  
Assignment 2  
Due 1/23/08

For all algorithm and data structure design problems please provide elegant pseudocode and an adequate explanation of your methods. If is often helpful to include small examples demonstrating the method. Put your name at the top of each sheet of paper that you turn in.

1. 2.1 page 50
2. 2.4 page 50. It suffices to show that  $\lim_{n \rightarrow \infty} \frac{\log^k n}{n} = 0$  which can be done using L'Hospital's rule from calculus.
3. Sorting algorithms can also be applied to lists. Suppose we have a list of records with two fields [key: integer, next: record pointer]. If  $x$  is a record, then  $x.key$  is the key value and  $x.next$  is the pointer to the next member of the list. For example, an unsorted list 5,3,1,4,10,7 is represented by 6 records, the first with 5, pointing to the second with 3, and so on. The last record has a pointer to null. Design pseudocode for a list version of insertion sort described at a high level by the following. For the empty list insertion sort returns the empty list. For a nonempty list, recursively insertion sort all but the first member of the list, then return the result of inserting the first member of the list properly into that result. You should define a recursive helper function that does the insertion into a sorted list. In the whole process the key field is never copied and the number of records remains the same. Give a recurrences for the worst-case running times of insertion sort and its helper function and solve them.
4. Continuing with lists we would like to implement recursive mergesort. There are two major steps: splitting a list into two equal (or almost equal) size lists and is merging two sorted lists. In the whole process the key field is never copied and the number of records remains the same.
  - (a) Design pseudocode that given a list returns a pair [first: record pointer, second: record pointer] where first and second point to two list of equal (or almost equal) size. Hint: if the list has zero, one, or two records, then this is easy to return the right pair of lists. Otherwise, recursively split the list, except for the first two records, then return the result of putting the first record at the front of the first list and the second record at the front of the second list.
  - (b) Design pseudocode that given two sorted lists returns the result of merging them into one sorted list.
  - (c) Use split and merge to define the pseudocode for list mergesort.