

CSE 326 Summer 2006

Assignment 6

Due: Wednesday, August 2

For all algorithm and data structure design problems, please provide elegant pseudocode and an adequate explanation of your methods. It 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. 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.
 - (a) 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.
 - (b) Give a recurrences for the worst-case running times of insertion sort and its helper function and solve them.
2. 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 it 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.

3. Suppose we have an unsorted array $A[1 \dots n]$ of integers with possible duplicates. Design a version of Quicksort that instead of partitioning into two sets, one whose elements are less than or equal to the pivot and a second whose elements are greater than or equal to the pivot, the new algorithm partitions into three sets, one whose elements are strictly less than the pivot, a second whose elements are strictly more than the pivot, and a third whose elements are equal to the pivot. Your algorithm should be in-place. One idea is that in the partitioning phase as we move the two pointers i and j toward each other we maintain the invariant that the array looks like:
- [elements equal to pivot] [elements less than pivot] [unknown elements] [elements greater than pivot] [elements equal to pivot]

When there are no unknown elements left then the elements can be rearranged to be of this form:

[elements less than pivot][elements equal to pivot][elements greater than pivot]

- (a) Design the Quicksort and Partition algorithms that implement this idea.
- (b) Show that your Quicksort algorithm runs in worst case time $O(dn)$ where d is the number of distinct keys in the array.
4. Suppose you are given as input n positive integers and a number k . Write an algorithm to determine if there are any four of them, repetitions allowed, that sum to k . Your algorithm should run in time $O(n^2 \log n)$. Partial credit will be given if your algorithm is correct but takes longer than $O(n^2 \log n)$. As an example, if $n = 7$, the input numbers are 6, 1, 7, 12, 5, 2, 14 and $k = 15$, the answer should be yes because $6+5+2+2 = 15$.

Hint #1: First solve the simpler problem that determines if there are any two numbers that sum to k .

Hint #2: The sum of four numbers is the sum of two pairs of numbers.