**CS 213**

**Practice Problems II**

1. ( SKIP Lists) Intuitively, it is easier to find an element that is nearby an element you’ve already seen. In a dynamic-set data structure, a finger search from x 􀀀 xto y \_ y is the following query: given the node in the data structure that stores the element x\_ , and given another element y\_ , find the node in the data structure that stores y \_ . Skip lists support fast finger searches in the following sense. Give an algorithm for finger searching from x \_ to y \_ in a skip list. Arrive at its complexity.

2. (Linked List) Let S and T be two sets of numbers, represented as unordered linked lists of distinct numbers. All you have are pointers to the heads of the lists, but you do not know the list lengths. Describe an O(min {|S| , |T|})-expected-time algorithm (*algorithmic steps are good enough – from which you can argue the complexity*) to determine whether S = T. You may assume that any operation on one or two numbers can be performed in constant time.

3. You are given a sequence of n numbers (positive or negative): . Your job is to select a subset of these numbers of maximum total sum, subject to the constraint that you can’t select two elements that are adjacent (that is, if you pick then you cannot pick either or ). Explain how you can find, in time polynomial in n, the subset of maximum total sum. ((*algorithmic steps are good enough – from which you can argue the complexity*).

4. Give a nonrecursive algorithm that performs an inorder tree walk.

5. What is the difference between the binary-search-tree property and the min-heap

property ? Can the min-heap property be used to print out the keys

of an n-node tree in sorted order in O(n) time? Show how, or explain why not.

6. Given a word w, write a program to find the words constructed out of the letters of w and which is lexico-graphically just before w.

7. Problems 8 , 16, 53, 58 in Sahni’s Book , Chapter 11.