HW # 4 Chapters 10, 11, 14 COP 3530 Fall 11
Instructor: Manu Sethi
TA in charge: Ravi Jampani

due Wednesday 11/16/2011
12:50 pm (in class)

Instructions: Submit via sakai.

1. Chapter 11 Binary Trees Exercise 50 Postorder Iterator
Write a class postOrderIterator as a public member of the class linked-
BinaryTree. Implement the methods hasNext() and next() in your iterator
class. Implement your class in the file linkedBinaryTree.h provided at the
book’s website.
Note that you will need to write an iterative code for this problem i.e.,
you need to iteratively traverse the data structure. Refer to the problem
statement in the book for the space and time constraints

2. Chapter 14 Binary Search Trees Exercise 14 Inorder Iterator

3. Hashing
Suppose we want to find the first occurrence of a string $P_1P_2......P_k$ in
a long input string $A_1A_2....A_N$. We can solve this problem by hashing
the pattern string, obtaining a hash value $H_p$ and comparing this value
with the hash value formed from $A_1A_2...A_k$, $A_2A_3...A_{k+1}$, $A_3A_4...A_{k+2}$,
and so on until $A_{N-k+1}A_{N-k+2}...A_N$. If we have a match of hash values,
we compare the strings character by character to verify the match. We
return the position (in $A$) if the strings actually do match, and we continue
in the unlikely event that the match is false.

(a) Show that if the hash value of $A_iA_{i+1}.....A_{i+k-1}$ is known, then the
hash value of $A_{i+1}A_{i+2}.....A_{i+k}$ can be computed in constant time.
(b) Show that the running time is $O(k+N)$ plus the time spent refuting
false matches.
(c) Write a C++ program to implement this algorithm. You are required
to make a hash table for this program.