S-Q 1998 - Assignment 3
Due 07/15/98 at 6:00pm
Last Date Accepted: 07/20/98 at 6:00pm

1 Getting Prepared
Copy all of the files from \~\libsq\assts\asst3 into your directory \~\q\asst3 (which should have been created when you ran q\setup for Assignment 1).

2 Exercises (100 points)
All work should be done in directory \~\q\asst3. The Makefile provided includes rules to build all the programs in this assignment; you shouldn't need to change it at all.
Answers to the written problems should be written out by hand.

1. (25 points) Complete all of the functions in tree_utils.c. Be sure to implement the functions according to the specifications in the comments in this file.
The functions in tree.c and tree_print.c are provided as helper functions, and tree_test.c is a simple test driver which you may modify in any way you see fit as you test your functions.
Hint- treeInOrder is more difficult than it might appear at first glance. Be sure that your algorithm is correct before you start to code it!

2. (25 points) Implement the game of animals by implementing function play Animals in file animals.c.
An animal tree is a binary tree where the internal nodes are called question nodes and represent questions, while the leaves are called animal nodes and represent the names of animals. The questions are always yes/no questions. All the left descendants of a question node correspond to a “yes” answer and all the right descendants a “no” answer. The game of animals is an interactive program that constructs an animal tree, based on input from the user, and uses this information to try to guess the identity of whatever animal the user thinks of.
Figure 1 shows an example dialog with the game, produced by sol_animals. (You may find it very useful to run sol_animals to get a feel for how your program should work. Do not feel compelled to provide a completely identical user interface, but do make sure that your program is easy to use and has a sensible user interface.) Figure 2 shows the corresponding animal tree that would exist at the end of the game illustrated in figure 1, and figure 3 shows how this tree looks as each new animal and question is added.
Many functions that you may find useful are already implemented in animals.c (with corresponding declarations and definitions in animals.h. You may use, modify, or ignore this code as you see fit.

3. (10 points) Starting with an empty B-tree of order 4, show what the B-tree would look like after inserting each of the following integer keys. (You should have one drawing per insertion, showing what the tree looks like after that insertion.)
10, 20, 1, 2, 3, 4, 11, 5, 12, 17, 14, 0

4. (15 points) Weiss, exercise 6.9, part a.
Remember that your goal is not to find the K smallest numbers—your goal is simply to find all the numbers smaller than X. You need only find the nodes in question — you do not need to delete them from the heap. (Note that it is possible that none of the numbers in the heap is smaller then X, and it is also possible that they all are.)
5. (25 points) Devise an algorithm that prints the keys in a binary tree in level-order, as described in Weiss, section 4.6. Hint—use a queue.

A level-order traversal processes the root node, then the keys of the nodes at depth 1 (from left to right), then the keys of the nodes at depth 2 (again from left to right), and so on.

For full credit, the traversal must run in time $O(n)$ and space $O(n)$, where $n$ is the number of nodes in the tree (regardless of the shape of the tree).

You do not need to implement your algorithm in C, but your algorithm should be detailed enough so that it would be easy to implement.

3 Graduate Problem (20 points)

We have defined the algorithms for pre-order, in-order, and post-order tree traversals recursively, but they can also be defined iteratively. Give algorithms for performing pre-order, in-order, and post-order tree traversals of a binary tree using only iterative constructs. (You may find it useful to begin by defining a single algorithm for binary tree traversals, and then show how this algorithm can be modified to perform each of these three traversals.)

Each of the traversals should require $O(n)$ time and $O(n)$ space, where $n$ is the number of nodes in the tree. You may use any of the ADTs or data structures you have seen in lecture or the reading. You must not assume anything specific about the implementation of the tree itself, and your algorithm must not destroy or alter any part of the tree.

Hint— the stack ADT will be very useful.

You do not need to implement your algorithm in C, but your algorithm should be detailed enough so that it would be easy to implement.

4 Submit Your Work

1. Use the submit program to hand in your work. The name of the course is cscisq, the assignment number is 3, and pathname of the files to submit is ~/Q/asst3, so the full command is:

   ```
   submit cscisq 3 ~/Q/asst3
   ```

   (See the UNIX tutorial for more info about submit.)

2. Print out a copy of your animals.c, tree_utils.c, and any other files you modify or create as part of your solution.
Hello, and welcome to the Game of Animals.

I don’t know the names of any animals.
Please tell me the name of an animal to help me get started:
\texttt{cat}

Would you like to play again? [yes/no] \underline{yes}
Think of an animal and I will guess it.
Is it a cat? [yes/no] \underline{yes}
I guessed it!

Would you like to play again? [yes/no] \underline{yes}
Think of an animal and I will guess it.
Is it a cat? [yes/no] \texttt{no}
I don’t know what animal it is. What is it?
\texttt{dog}
Please tell me a question that I can ask to distinguish a cat from a dog.
The answer should be "yes" for a dog and "no" for a cat.
\textbf{Does it bark?}
Thank you. I now know how to tell a cat from a dog.

Would you like to play again? [yes/no] \underline{yes}
Think of an animal and I will guess it.
Does it bark? [yes/no] \texttt{no}
Is it a cat? [yes/no] \texttt{no}
I don’t know what animal it is. What is it?
\texttt{fish}
Please tell me a question that I can ask to distinguish a cat from a fish.
The answer should be "yes" for a fish and "no" for a cat.
\textbf{Does it live in the water?}
Thank you. I now know how to tell a cat from a fish.

Would you like to play again? [yes/no] \texttt{no}
Goodbye.
Does it bark?

yes

no

dog

Does it live in the water?

yes

no

fish

cat

Figure 2: An animal tree.

Figure 3: Construction of the animal tree.