CS 2123
Summer 2014

Homework 6

Assigned: 7/21/2014
Due: 7/28/2014 11:59 pm

1. Dynamic Sequential Insertion Search (25 pts)
Write a C program that Implements a sequential insertion search using dynamic memory allocation.
You MUST use calloc, realloc, and pointer arithmetic. You MAY NOT use a statically declared array nor bracket access notation.
Initially allocate and initialize space for **10** integers.
Read a file of integers.
For each integer read from the file, perform a sequential insertion search.
If the search is successful, **write to an output file, named output.txt:** "SEARCH RESULT: <value> is at <key>", where <value> is the integer from the file and <key> is the array index.
Otherwise, write to the output file: "INSERTING <value> at <key>".
Once you fill your array, you must reallocate space for **5** more elements in your array.
Don't forget to initialize the new elements!
Continue reading the file, searching and storing the integers, and writing to the output file.
Repeat reallocation, reading, searching, and storing until the entire file has been read.

The format of the data file is:
<int>
<int>
...

2. Verifying Binary Search Tree Performance (30 pts)
Exercise 7.2.3. in the textbook.
You will need to write a C program to do this. Use the search and insertion algorithm on page 404.
Use the random number generator from Homework 4, but generate numbers with a range from 1 to 10,000.
Test your tree search performance with at least 100 randomly generated integers.
To verify performance you need to keep track of the # of comparisons (i.e., # of node visits) for each search and the the # of nodes in your binary tree when each search is made.
Present your results in a chart or table, along with values for the boundary function $O(\log_2 n)$. Have at least 10 different values of *n* (i.e., 10 different tree sizes ranging from 5 to 100 nodes). Run at least 10 search/insertions per tree size (i.e., value of n) and

calculate an average # of comparisons for your chart.

For example:

n	**Average # comparisons/visits in my search**	**$\log_2 n$**
5	3.49	2.32
10	4.02	3.32
15	5.11	3.91
20	5.79	4.32
25	6.35	4.64
…		

3. Understanding Multiway Search Trees (45 pts)
Exercise 7.3.2. in the textbook.
You only have to produce Top-down and B trees. You do not have to produce B+ nor Digital Search Trees.
Ignore case and omit duplicate words.
Instead of choosing your own, use the following paragraph (paraphrased from Brooks' The Mythical Man Month):

"Of all the monsters who fill the nightmares of our folklore, none terrify more than werewolves, because they transform unexpectedly from the familiar into horrors. For these, we seek bullets of silver that can magically lay them to rest. The familiar software project has something of this character, usually innocent and straightforward, but capable of becoming a monster of missed schedules, blown budgets, and flawed products. So we hear desperate cries for a silver bullet, something to make software costs drop as rapidly as computer hardware costs do."

Deliverables:
The answers to homework problem 3 should be typed/drawn and submitted in a single document (PDF).
Problems 1 and 2 should be submitted as separate C source code files ALONG WITH your output file for problem 1 (called output.txt). I need all of the C source that you produce for this assignment problems so that I may compile and execute it.

Archive and submit the files to UTSA's Blackboard system (utsa.blackboard.edu)