EECS 281 Fall 2007
Homework Assignment 2
Due Thursday 10/11/07 at 1:00 PM

Instructions: Complete all problems and turn in a hard copy to the EECS 281 lock box in 2420 EECS by the specified due date. Type or write legibly and include both your name and uniqname on each page of the assignment. There are 7 questions worth a total of 60 points.

1.) (8 points) Suppose that you open a savings bank account with an APR of 6.25% and make a $2,000 initial deposit. In addition to that, you also deposit $1,000 after the first year with the amount incrementing by $100 for every consecutive year after the first. For example, the deposit schedule for first three years is:

<table>
<thead>
<tr>
<th>Year</th>
<th>Deposit Amount ($)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1000</td>
</tr>
<tr>
<td>2</td>
<td>1100</td>
</tr>
<tr>
<td>3</td>
<td>1200</td>
</tr>
<tr>
<td></td>
<td>...</td>
</tr>
</tbody>
</table>

a.) (4 points) Derive a recurrence relation that would give the balance in the account after n years (Provide a logical reasoning for partial credit)

b.) (4 points) Write a recursive function int getBalance(int noOfYears) that would return the balance of the account after noOfYears

2.) (9 points) Consider the following code:

```c
void mysteryFunction(int* array, int low, int high)
{
    int a=low;
    int b=high;
    int c;

    int mid = (low+high)/2;
    int x=array[mid];

    do
    {
        while(array[a] < x)
            a++;
        while(array[b] > x)
```
b--; 
if(a <= b) 
{
    c = array[a];
    array[a] = array[b];
    array[b] = c;
    a++;
    b--;
} 
} while (a <= b);

if(low < b) 
mysteryFunction(array, low, b);
if(a < high) 
mysteryFunction(array, a, high);

a.) (3 points) What is the best case running time of the function?
b.) (3 points) What is the average case running time of the function?
c.) (3 points) What is the worst case running time of the function?

3.) (8 points) A binary tree can be represented in several ways. One way of representing binary trees is through lists. Here is a list representation of a simple binary tree (the first element is the root, the second element is the left subtree, and the third element is the right subtree):

(a (b (d (k () (i)) ()) (e () ())) (c (f () ()) (g () ()))))

a.) (2 points) Draw a tree diagram of the binary tree represented above
b.) (2 points) Run a depth-first search on the binary tree and print out the nodes in the order of completion of traversal
c.) (4 points) Develop a function called runDFS (char* myArray) that would print out the same output as in part b). myArray is a array of characters as shown in the list representation above (Feel free to add parameters if you feel that they are necessary).

4.) (9 points) Draw the frequency array and Huffman tree for the following string:

"can we say dogs do have to not not spot hot post or cats but i love eecs at michigan"

5.) (9 points) Draw a standard trie for the following set of strings:
6.) (9 points) Draw a fully-expanded n-ary expression tree for each of the following:

   a.) (3 points) Binary: $5^a b + ((3/7) - (2+16))/a - 14/2$

   b.) (3 points) N-ary: $f(g(r(x, y, z), m(a, b), c), h(n(0, 14, g(c, d))))$

   c.) (3 points) Binary: $Z(10)$, where $Z$ is defined by the recurrence relation below:

   
   \[
   Z(x) = \begin{cases} 
   2 & \text{for } n=0, \\
   4 & \text{for } n=1, \\
   n^3 - Z(n-1) \times Z(n-2) & \text{for } n>1 
   \end{cases}
   \]

7.) (8 points) Consider the following list of words:
{cucumber, salad, carrot, aspirin, sausage, watermelon, fruitcake, applesauce, einstein, jamin, elephant, carrot, cookie, fruitcake}

   a.) (4 points) Draw the resulting binary search tree if we insert the words into the tree in the order given above. Assume the standard alphabetical (A-Z) ordering. Compute the height of the tree.

   b.) (4 points) For an unbalanced search tree, what is the best-case time complexity, if the tree has $n$ words? What is the worst case?