May 6, 2003       CS372 Final Exam       Time: 120 min

Instructions: The exam has eleven pages. Show intermediate steps, if any, in your solutions if you want partial credit.

Suggestion: Read all problems and their weightages carefully before answering any of them.

1. Consider the following algorithm:

Algorithm $\text{Test}(\text{int } n)$
   if $(n > 1)$ then
      $m \leftarrow 1$
      while $(m + m < n)$ do
         $m \leftarrow m + 1$
         $\text{Test}(m)$
         $\text{Test}(n - m)$

(a) Let $G(n)$ denote the running time of $\text{Test}$ on an input $n > 0$. Derive a recurrence relation for $G(n)$.

6 pts
(b) Determine explicitly a function $f(n)$ such that $G(n)$ is $\Theta(f(n))$.  

4 pts
2. Assume that you have a binary search tree where each node of the tree has five fields—
parent, left, right, key and count. The first four fields have obvious meanings. The
field count for node \( v \) is supposed to keep track of the number of nodes in the subtree
rooted at \( v \) (including \( v \) itself). Assume that you have a tree \( T \) where the count values
have not been computed. Give pseudocode for a simple recursive procedure COUNTNODES
that given a pointer to the root of binary search tree \( T \) sets the count field correctly for
each node in \( T \). Analyze the worst-case running time of your procedure.

\[ 5 \text{ pts} \]
3. Assume that you have a binary search tree $T$ as in problem 2 but where the count field for each node in $T$ has been correctly set. Give pseudocode for a simple recursive procedure RANK that given two arguments – a pointer to the root of binary search tree $T$ and an integer $k$ – returns the $k^{th}$ smallest key in $T$. Analyze the worst-case running time of your procedure.

$10$ pts
4. Suppose a file $F$ has six characters \{a, b, c, d, e, f\} with the frequencies given below:

\begin{tabular}{ccccc}
character & a & b & c & d & e & f \\
frequency  & 1 & 2 & 4 & 8 & 16 & 32 \\
\end{tabular}

(a) Draw the code tree constructed by the Huffman’s Algorithm for the above input.

5 pts
(b) Calculate the number of bits in the resulting encoded file when this code is used for encoding $F$.

3 pts
5. Draw a Red-Black Tree that has 11 internal (non-dummy) nodes and has greatest depth among all Red-Black Trees with 11 internal nodes.

7 pts
6. The computer corporation MacroHard has a company hierarchy consisting of \( n+1 \) levels \( \{0, 1, 2, \ldots, n\} \). The company hires new employees only at the lowest level (level 0) and promotes them to higher levels depending on how hard they work. However, a single promotion in the company promotes an employee to a level at most 3 levels higher than the previous level, i.e., an employee at level \( i \) can be promoted only to levels \( i+1, i+2 \) or \( i+3 \). Getting faster promotions requires more effort. The goal of this problem is to develop an algorithm that computes a way to rise to the top of a MacroHard company hierarchy with least effort.

\((i)\) Suppose the MacroHard company hierarchy has six levels \( \{0, 1, 2, \ldots, 5\} \). The efforts required to get direct promotion from level \( i \) to level \( i+t \) is represented by the edge weights in the graph below as well as represented as a table \( C \).

\[
\begin{align*}
\text{Levels / } t & \quad 1 & 2 & 3 \\
0 & 10 & 28 & 57 \\
1 & 20 & 48 & 80 \\
2 & 30 & 50 & 100 \\
3 & 40 & 60 & - \\
4 & 45 & - & - \\
5 & - & - & - 
\end{align*}
\]

Table \( C \) of required efforts.
\( C(i, t) = \text{Effort required to get promoted from level } i \text{ to level } i+t \text{ directly.} \)
(a) What is the minimum total effort required to rise from level 0 to level 4? What sequence of promotions requires this minimum effort? \hspace{2cm} 3 pts

(b) What is the minimum effort required to rise from level 0 to level 5? What sequence of promotions requires this minimum effort? \hspace{2cm} 5 pts
(ii) In general, assume that you have $n + 1$ levels $0, 1 \ldots n$. Let $C[i, t]$ denote the effort required to get a direct promotion from level $i$ to $i + t$ where $1 \leq t \leq 3$. For $i \geq 1$, let $E(i)$ denote the minimum total effort required to rise from level 0 to level $i$. Derive a recurrence relation for $E(i)$.

**Hint:** Suppose your last promotion to get to level $i$ is a $t$-level promotion from level $i - t$ to level $i$. Then you have spent effort in rising from position 0 to $i - t$ before and you require effort $C[i - t, t]$ for your last promotion.

*10 pts*
(iii) Use the recurrence relation in (ii) to provide pseudocode for a dynamic programming algorithm that computes the minimum effort required to rise from level 0 to level $n$ in the MacroHard Hierarchy. Analyze the running time of your algorithm.  

12 pts

Extra Credit: Modify your algorithm to compute not only the total minimum effort but also to print out a sequence of promotions that requires the total minimum effort.  

6 pts