Final Exam

(3 hours open book exam)

<table>
<thead>
<tr>
<th>Problem</th>
<th>credit</th>
<th>maximum</th>
</tr>
</thead>
<tbody>
<tr>
<td>Problem 1</td>
<td></td>
<td>10</td>
</tr>
<tr>
<td>Problem 2</td>
<td></td>
<td>10</td>
</tr>
<tr>
<td>Problem 3</td>
<td></td>
<td>10</td>
</tr>
<tr>
<td>Problem 4</td>
<td></td>
<td>10</td>
</tr>
<tr>
<td>Problem 5</td>
<td></td>
<td>10</td>
</tr>
<tr>
<td>Problem 6</td>
<td></td>
<td>10</td>
</tr>
<tr>
<td>Problem 7</td>
<td></td>
<td>10</td>
</tr>
<tr>
<td>Total</td>
<td></td>
<td>70</td>
</tr>
</tbody>
</table>

Name: __________________________
Name: ________________________

**Problem 1.** (10 = 3 + 4 + 3 points). Recall that the weighted external path length of a tree is the sum of weighted depths of all external nodes.

(a) Is it true that the weighted external path length of the Huffman tree is always smaller than or equal to that of the optimal weighted binary search tree that stores the keys (in sorted order) in its external nodes? Justify your answer.

(b) Recall that the order of the two children of an internal node in a Huffman tree is not important. How many orderings of the external node can you generate by switching left and right subtrees at internal nodes?

(c) Can we use the Huffman tree to efficiently search in a collection of weighted keys? Why or why not?
**Problem 2.** (10 = 2 + 2 + 2 + 2 + 2 points). Each of the following five statements is true (T) or false (F) or not known to be true or false (NK). A correct answer counts as 2 points, an incorrect answer counts as −1 point, and no answer counts as 0 points. Giving more than one answer to the same question is considered an incorrect answer and worth −1 point.

Let $G$ be an undirected graph and $k$ a positive integer.

(a) To decide whether or not $G$ has an independent set of size $k$ is NP-hard.

(b) There can be no polynomial time algorithm that decides whether $G$ has an independent set of size $k$.

(c) To decide whether or not $G$ has an independent set of size 5 is NP-hard.

(d) Suppose you find a polynomial time algorithm that determines whether $G$ has an independent set of size $k$ or larger. This implies $\text{NP} = \text{co-NP} = \text{P}$.

(e) Suppose you prove $\text{INDEPENDENT SET} \in \text{co-NP}$. This implies $\text{NP} = \text{co-NP} = \text{P}$.
Name: _______________________

**Problem 3.** (10 points). Let $A[0..n-1]$ store $n$ items and let $0 \leq k \leq n - 1$ be an integer. Write an algorithm that takes $O(n)$ time to rotate $A$ by $k$ positions, as shown in Figure 1. Do the

\[
\begin{array}{c|c|c}
0 & 1 & n-1 \\
A & B & Z
\end{array}
\quad\rightarrow\quad
\begin{array}{c|c|c|c|c|c|c}
0 & k-1 & k & k+1 & n-1 \\
& & & & & \\
& & & & & \\
\end{array}
\]

Figure 1: Array of $n$ elements rotated by $k$ positions.

rotation in-place (that is, only a constant number of extra variables are permitted). Describe your algorithm in words and justify it.
Problem 4. (10 = 2+2+2+2+2 points). Consider a binary tree with height \( \ell \) and \( n = 2^\ell \) leaves. The left and right children of a node \( \mu \) are denoted by \( \mu \rightarrow \ell \) and \( \mu \rightarrow r \). The following generic algorithm starts at the root and visits nodes depending on functions \( f \) and \( g \).

```c
void VISIT(\mu)
    if \( \mu \neq \text{NULL} \) then
        VISIT(f(\mu)); VISIT(g(\mu))
    endif.
```

Determine the running time (in big-Oh notation) for the following specifications of \( f \) and \( g \).

(a) \( f(\mu) = \mu \rightarrow \ell, g(\mu) = \mu \rightarrow r \).
(b) \( f(\mu) = \mu \rightarrow \ell \rightarrow \ell, g(\mu) = \mu \rightarrow r \rightarrow r \).
(c) \( f(\mu) = \mu \rightarrow \ell, g(\mu) = \text{NULL} \).
(d) \( f(\mu) = \mu \rightarrow \ell \rightarrow \ell, g(\mu) = \text{NULL} \).
(e) \( f(\mu) = \mu, g(\mu) = \text{NULL} \).
Problem 5. (10 points). Suppose you are given the adjacency lists representation of an undirected graph, consisting of a linear array $V[1..n]$ for the vertices and a linked list $V[i].adj$ of neighbors for each vertex $i$. Let $2m$ be the total length of the lists. Give an $O(n + m)$ time algorithm for checking whether the representation contains multi-edges. (Recall that a graph contains multi-edges if it contains two or more edges connecting the same two vertices.) Describe your algorithm in words and justify it.
Problem 6. (2 + 3 + 5 points). Consider the square grid of points \((i, j)\) with \(0 \leq i \leq 2m\) and \(0 \leq j \leq 2n\), as shown for \(m = 6\) and \(n = 4\) in Figure 2. A maximal monotone path begins at \(A = (0, 0)\) and reaches \(Z = (2m, 2n)\) by an intermixed sequence of \(2m\) right and \(2n\) up moves.

(a) For \(2m = 2\) and \(2n = 4\) there are 15 maximal monotone paths. Draw all and count how many pass through or below the point \(M = (m, n)\) in the center.

(b) How many maximal monotone paths are there for general values of \(m\) and \(n\)? Give a closed form expression for the result.

(c) How many maximal monotone paths pass through or below the point \(M\)? Give a closed form expression for the result.
Problem 7. (10 = 2 + 4 + 4 points). Let $B$ be a full binary tree with $n$ external nodes. (Recall that in a full binary tree every internal node has exactly two children.) Define the right-depth of an external node $\lambda$ as the number of right edges on the path from the root to $\lambda$. Define the external right path length, $r(B)$, as the sum of right-depths over all external nodes.

(a) Which tree with $n$ external nodes minimizes $r(B)$ and which such tree maximizes $r(B)$?

(b) Suppose that for every internal node of $B$ there are at least as many external nodes in the left as there are in the right subtree. Prove that $r(B) \leq n \log_2 n$.

(c) Suppose the running time of some algorithm is $T(n) = T(k) + T(n - k) + \min\{k, n - k\}$, where $k$ can be any integer between 1 and $n - 1$. Prove that $T(n) = O(n \log n)$. 