# Midterm

(April 26, 2001)

## Note

This is a closed-book exam. Each problem accounts for 10 points, unless otherwise marked.

## Problems

1. Find an open Gray code of length $\lceil \log_2 13 \rceil$ $(= 4)$ for 13 objects. Show how the Gray code is constructed systematically from Gray codes of smaller lengths.

2. Let $a_1, a_2, \cdots, a_n$ be positive real numbers such that $a_1 a_2 \cdots a_n = 1$. Prove *by induction* that $(1 + a_1)(1 + a_2) \cdots (1 + a_n) \geq 2^n$. (Hint: In the inductive step, try introducing a new variable that replaces two chosen numbers from the sequence.)

3. Below is an algorithm for solving a variant of the Towers of Hanoi puzzle with an additional fourth peg $D$; `Towers_Hanoi` is an algorithm for the original puzzle.

```
Algorithm Four_Towers_Hanoi(A,B,C,D,n);
begin
   if n<=2 then
      Towers_Hanoi(A,B,C,n);
   else
      Four_Towers_Hanoi(A,D,B,C,n-2);
      Towers_Hanoi(A,B,C,2);
      Four_Towers_Hanoi(D,B,C,A,n-2);
end;
```

   Let $T(n)$ denote the number of moves needed for $n$ disks. Write a recurrence relation for $T(n)$ and solve it.

4. Show all intermediate and the final AVL trees formed by inserting the numbers 1, 7, 2, 6, 3, 5, and 4 (in this order). If a rotation is performed during an insertion, please also show the tree before the rotation. (15 points)

5. The Knapsack Problem is defined as follows: Given a set $S$ of $n$ items, where the $i$th item has an integer size $S[i]$, and an integer $K$, find a subset of the items whose sizes sum to exactly $K$ or determine that no such subset exists.

Below is an algorithm for determining whether a solution to the problem exists.

```
Algorithm Knapsack $(S, K)$;
begin
    $P[0,0].exist := true$;
    for $k := 1$ to $K$ do
        $P[0,k].exist := false$;
    for $i := 1$ to $n$ do
        for $k := 0$ to $K$ do
            $P[i,k].exist := false$;
            if $P[i-1,k].exist$ then
               $P[i,k].exist := true$;
               $P[i,k].belong := false$
            else if $k - S[i] \geq 0$ then
                    if $P[i-1,k-S[i]].exist$ then
                       $P[i,k].exist := true$;
                       $P[i,k].belong := true$
end
```

(a) Modify the algorithm to solve a variation of the knapsack problem where each item has an unlimited supply. In your algorithm, please change the type of $P[i,k].belong$ into integer and use it to record the number of copies of item $i$ needed. (5 points)

(b) Design an algorithm to recover the solution recorded in the array $P$ of the algorithm in (a). (10 points)

6. Given as input two sorted arrays $A$ and $B$, each of $n$ numbers (in an increasing order), and another number $x$, design an algorithm with running time $O(n)$ to determine whether there exist an element in $A$ and an element in $B$ whose sum is exactly $x$. (Hint: Recall the ideas of the $O(n)$ soluton to the Celebrity Problem discussed in class.)

7. Apply the quicksort algorithm to the following array. Show the contents of the array after each partition operation.

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 5 | 1 | 8 | 11 | 2 | 12 | 7 | 3 | 6 | 10 | 4 | 9 |

8. Below is a variation of the $n$-coins problem.

   > You are given a set of $n$ coins $\{c_1, c_2, \ldots, c_n\}$, among which at least $n-1$ are identical "true" coins and at most one coin is "false". A false coin is *lighter* or *heavier* than a true coin. Also, you are given a balance scale, which you may use to compare the total weight of any $m$ coins with that of any other $m$ coins. The problem is to find the "false" coin, or show that there is no such coin, by making some sequence of comparisons using the balance scale.

   Show that in the worst case it is impossible to solve the $n$-coins problem with $k$ comparisons (for any $n$ and $k$) if $n > \frac{(3^k-1)}{2}$. (Hint: Think about decision trees and how many possible outcomes there can be for the problem.)

9. Draw a Huffman tree for a text that contains eight characters $A$, $B$, $C$, $D$, $E$, $F$, $G$, and $H$ with frequencies 8, 2, 5, 6, 14, 3, 2, and 4, respectively.