Midterm

(April 25, 2002)

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 $\lceil \log_2 15 \rceil$ $(= 4)$ for 15 objects. Show how the Gray code is constructed systematically from Gray codes of smaller lengths.

2. What is wrong with the following proof?

   Claim: In any non-empty set of horses, all horses are of the same color.

   Proof: By induction on the size n of a set.
   Base case: $n = 1$, there is just one horse and it has the same color as its own.
   Inductive step: $n = k+1, k \geq 1$. Consider any set H of size $k+1$. Remove one horse h from H to get H_1 of size k. From the induction hypothesis, all horses in H_1 are of the same color. Put the removed horse h back and remove a different horse from H to get H_2 of size k. Again, from the induction hypothesis, all horses in H_2 are of the same color. H_2 contains horse h and some other horses from H_1. It follows that horse h has the same color as those in H_1 and, therefore, all horses in H are of the same color.

3. Below is a theorem from Manber's book:

   > For all constants $c > 0$ and $a > 1$, and for all monotonically increasing functions $f(n)$, we have $(f(n))^c = O(a^{f(n)})$.

   Prove, by using the above theorem, that $n^2(\log n)^4 = O(n^{2.5})$.

4. Show all intermediate and the final AVL trees formed by inserting the numbers 2, 4, 5, 8, 7, 6, 3, and 1 (in this order). Please use the following ordering convention: the key of an internal node is larger than that of its left child and smaller than that of its right child. If a rotation is performed during an insertion, please also show the tree before the rotation.

5. Suppose that you are given an algorithm as a *black box* (you cannot see how it is designed) that has the following properties: If you input any sequence of real numbers and an integer k, the algorithm will answer "yes" or "no," indicating whether there is a subset of the numbers whose sum is exactly k. Show how to use this black box to find the subset whose sum is k, if it exists. You should use the black box $O(n)$ times (where n is the size of the sequence).

6. The Knapsack Problem is defined as follows: Given a set S of n items, where the ith 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
   $\quad P[0,0].exist := true$;
   $\quad$ **for** $k := 1$ **to** K **do**
   $\quad\quad P[0,k].exist := false$;
   $\quad$ **for** $i := 1$ **to** n **do**
   $\quad\quad$ **for** $k := 0$ **to** K **do**
   $\quad\quad\quad P[i,k].exist := false$;
   $\quad\quad\quad$ **if** $P[i-1,k].exist$ **then**
   $\quad\quad\quad\quad P[i,k].exist := true$;
   $\quad\quad\quad\quad P[i,k].belong := false$
   $\quad\quad\quad$ **else if** $k - S[i] \geq 0$ **then**
   $\quad\quad\quad\quad\quad$ **if** $P[i-1, k-S[i]].exist$ **then**
   $\quad\quad\quad\quad\quad\quad P[i,k].exist := true$;
   $\quad\quad\quad\quad\quad\quad P[i,k].belong := true$
   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.

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

7. 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.)

8. Rearrange the following array into a (max) heap using the bottom-up approach.

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
3	2	5	11	10	14	7	6	8	1	13	4	15	12	9

   Show the result after each element is added to the part of array that already satisfies the heap property.

9. Draw a Huffman tree for a text with the following frequency distribution: $A:9$, $B:3$, $C:6$, $D:5$, $E:16$, $F:4$, $G:2$, and $H:1$.

10. Compute the *next* table as in the KMP algorithm for the string *aababaabaab*. Please show the details of how $next[11]$ is computed using $next[1..10]$.