Midterm

Note

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

Problems

1. Construct an open gray code of length $\lceil \log_2 19 \rceil$ $(= 5)$ for 19 objects. Please describe how the gray code is constructed *systematically* from gray codes of smaller lengths.

2. Consider the following program segment in the celebrity algorithm.

```
i := 1;
j := 2;
next := 3;
while next <= n+1 do
   if Know[i,j] then i:= next
   else j := next;
   next := next + 1;
   end;
if i = n+1 then candidate := j
else candidate := i;
```

   (a) Find an appropriate loop invariant for the while loop that is sufficient to show that `candidate` will be the only possible candidate for the celebrity after the execution of the segment.
   (b) Prove that the loop invariant found above is indeed a loop invariant.

3. Find the asymptotic behavior of the function $T(n)$ defined as follows:

$$\begin{cases} T(1) = 1 \\ T(n) = 4T(n/2) + n^2, \quad n = 2^i \ (i \geq 1) \end{cases}$$

   You should try to solve this problem without resorting to the general theorem for divide-and-conquer relations discussed in class. The asymptotic bound should be as tight as possible. (Hint: guess and verify by induction.)

4. Show all intermediate and the final AVL trees formed by inserting the numbers 9, 7, 1, 0, 2, 5, 8, 4, 6, and 3 (in this order) into an empty tree. 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 re-balancing operations are performed, please also show the tree before re-balancing and indicate what type of rotation is used in the re-balancing.

5. Consider solutions to the union-find problem discussed in class. Suppose we start with a collection of ten elements: A, B, C, D, E, F, G, H, I, and J.

   (a) Assuming the balancing, but not path compression, technique is used, draw a diagram showing the grouping of these ten elements after the following operations are completed:

      i. union(A,B)
      ii. union(C,D)
      iii. union(E,F)
      iv. union(G,H)
      v. union(I,J)
      vi. union(A,D)
      vii. union(F,G)
      viii. union(D,J)
      ix. union(D,H)

      In the case of combining two groups of the same size, please always point the second group to the first.

   (b) Repeat the above, but with both balancing and path compression.

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
   $P[0,0].exist := true$;
   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) Design an algorithm to recover the solution recorded in the array P. (5 points)

(b) Modify the given algorithm to solve a variation of the knapsack problem where each item has an unlimited supply. (10 points)

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

Please show the result (i.e., the contents of the array) after a new element is added to the current collection of heaps (at the bottom) until the entire array has become a heap.

8. Prove that the sum of the heights of all nodes in a complete binary tree with n nodes is at most $n - 1$. (A complete binary tree with n nodes is one that can be compactly represented by an array A of size n, where the root is stored in $A[1]$ and the left and the right children of $A[i]$, $1 \leq i \leq \lfloor \frac{n}{2} \rfloor$, are stored respectively in $A[2i]$ and $A[2i+1]$. Notice that, in Manber's book a complete binary tree is referred to as a balanced binary tree and a full binary tree as a complete binary tree. Manber's definitions seem to be less frequently used. Do not let the different names confuse you.)

9. Let $x_1, x_2, \cdots, x_{2n-1}, x_{2n}$ be a sequence of $2n$ real numbers. Design an algorithm to partition the numbers into n pairs such that the maximum of the n sums of pair is minimized. It may be intuitively easy to get a correct solution. You must explain how the algorithm can be designed using induction. (15 points)