(1) Read the following data in the given order, and show the corresponding trees.

7, 8, 9, 2, 1, 5, 3, 6, 4

(a) Binary search tree (5%)

(b) AVL tree (5%)

(c) 2-3 tree (5%)

(2) Assume a document is composed of several chapters, and a chapter is divided into several sections. Now we have a huge document collection. Design a data structure that can support the following queries efficiently.

(i) Retrieve those sections that contain a specific word.

(ii) Retrieve those chapters that contain a specific word.

(iii) Retrieve those documents that contain a specific word.

Specify how the data structure you proposed can work efficiently for the above queries. (15%)

(3) Consider the following procedure to insert *item* into a min-max heap of size *n*.

```
void min_max_insert(element heap[], int *n, element item)
{    int parent;
     (*n) ++;
     parent = (*n) / 2;
     if (!parent) heap[1] = item;
     else switch(level(parent)) {
          case FALSE:
               if (item.key < heap[parent].key) {
                    heap[*n] = heap[parent];
                    verify_min(heap, parent, item); }
               else verify_max(heap, *n, item);
               break;
          case TRUE:
               if (item.key > heap[parent].key) {
                    heap[*n] = heap[parent];
                    verify_max(heap, parent, item); }
               else verify_min(heap, *n, item);
}}
```

(a) Write *verify_min* function in connection with the above procedure. (10%)

(b) Where is the maximum key in a min-max heap of size n ($n > 1$)? Show how to reconstruct the heap when the maximum key is deleted. (10%)

接背面

(4) Let A[1..n] be an array of n distinct numbers. If i<j and A[i]>A[j], then the pair (i,j) is called an inversion of A.

(a) List the five inversions of the array <2,3,7,6,1>. (5%)

(b) What array with elements from the set {1,2,…,n} has the most inversions? How many does it have? (5%)

(c) Give an algorithm that determines the number of inversions in any permutation on n distinct numbers in O(n log n) worst-case time. (Hint: Modify merge sort.) (15%)

(5) What is an optimal Huffman code for the following set of frequencies, based on the first 7 Fibonacci numbers?

a:1 b:1 c:2 d:3 e:5 f:8 g:13

Can you generalize your answer to find an optimal code when the frequencies are the first n Fibonacci numbers? (10%)

(6) Give asymptotically tight upper (big O) bounds for T(n) in each of the following recurrences. (15%)

(a) $T(n)=T(n/5)+T(7n/10)+n$

(b) $T(n)=T(n/3)+T(2n/3)+n$

(c) $T(n)=n+(4/n)(T(1)+T(2)+\ldots+T(n-1))$