浙江大学 2005－2006 学年冬季学期

《高级数据结构与算法》课程期末考试试卷

开课学院：_软件学院_ ，考试形式：闭卷，允许带___入场

考试时间：_2005_年_1_月_10_日,所需时间：_120_分钟

考生姓名：__________学号：____________专业：__________教师：__________

题序	一	二	三	四	总 分
得分					
评卷人					

NOTE: Please write your answers on the answer sheet.
注意：请将答案填写在答题纸上。

I. Please fill in the blanks (There could be multiple answers for one blank). (30 points) ***Note: <u>Zero point</u> for a blank selection since there is at least one answer for each problem.***

(1) For an AVL tree, ______ is(are) correct. (2 points)

a. The height of left and right sub-trees differs by at most 1.
b. The *insertion* operation can be performed in $O(\log N)$ time.
c. An AVL tree is a kind of binary search tree.
d. The purpose of using AVL tree is to save some space.

(2) An AVL tree of height 6 has the minimum size of ____ nodes. (2 points)

a. 31　　b. 32　　c. 33　　d. 34

(3) For a B-tree, ______ is(are) correct. (2 points)

a. A B-tree is not a search tree.
b. All the leaves have the same depth.
c. All non-leaf nodes of a 2-3 tree have 1 or 3 children.
d. Generally a B-tree is not a binary tree.

(4) For a leftist heap, ______ is(are) **NOT** correct. (2 points)

a. The total number of nodes in the left sub-tree is always no less than that in the right sub-tree.
b. A leftist heap of N nodes has a right path containing at most $\lfloor \log(N+1) \rfloor$ nodes.
c. Comparing to binary heaps, leftist heaps are more suitable to do *merge*.
d. A binary heap is also a leftist heap.

(5) For a skew heap,______ is(are) **NOT** correct. (2 points)

a. Skew heap is a kind of leftist heap.

b. The right path of a skew heap can be arbitrarily long.

c. Comparing to leftist heaps, skew heaps are always more efficient in running time for every *merge*.

d. Comparing to leftist heaps, skew heaps are always more efficient in space.

(6) Among the following statements, ______is(are) correct. (2 points)

a. The amortized time is an average time of N successive operations.

b. The amortized time is the worst-case time of N successive operations in total.

c. The potential function of N successive insertions of a binomial queue is the number of trees.

d. The amortized time could be much more than the actual time for one operation.

(7) For the followings, the $O(\log N)$ case(s) is(are)_____, the $O(N)$ case(s) is(are)_____. (2 points)

a. The worst-case running time of merging two leftist heaps.

b. The worst-case running time of merging two skew heaps.

c. The amortized cost per *merge* of a skew heap.

d. Building a leftist heap from N numbers.

(8) For the following problems, ___is(are)undecidable problem(s)and ____ is(are) NP-complete problem(s). (3 points)

a. Euler circuit problem

b. Hamiltonian cycle problem

c. Satisfiability problem

d. Halting problem

e. All-pairs shortest paths problem

f. Bin packing problem

(9) For the following statements, ____ is(are) **NOT** correct. (2 points)

a. All NP problems are decidable.

b. All NP-complete problems are NP problems.

c. All decidable problems are NP problems.

d. All NP problems can be solved in polynomial time in a non-deterministic machine.

(10) For a binomial queue containing 23 nodes, please give the names of non-empty binomial trees in this binomial queue (e.g. B0, B1, B3): ______ . (5 points)

(11) The turnpike reconstruction problem is to reconstruct a point set from distances between every pair of points. Given a set of distances { 1, 2, 3, 3, 4, 5, 7, 7, 8, 10 }, the corresponding point set contains __points with coordinates _____________. (6 points)

II. Given the function descriptions of the following two programs, please fill in the blank lines of code. (15 points)

(1) A single rotation between node K1 and its right child in an AVL tree. (9 points)

```
struct  AvlNode
{
  ElementType  Element;
  AvlTree Left;
  AvlTree Right;
  int Height;
}
Typedef struct AvlNode *Position;
Typedef struct AvlNode *AvlTree;
```

```
static Position
SingleRotateWithRight ( Position K1 )
{
    Position K2;
    ①________________________________;
    ②________________________________;
    ③________________________________;
    K1->Height =
    Max(Height(K1->Left),Height(K1->Right))+1;
    K2->Height =
    Max(Height(K2->Right),K1->Height)+1;
    return K2;
}
```

(2) Finding the optimal ordering of matrix multiplications. (6 points)

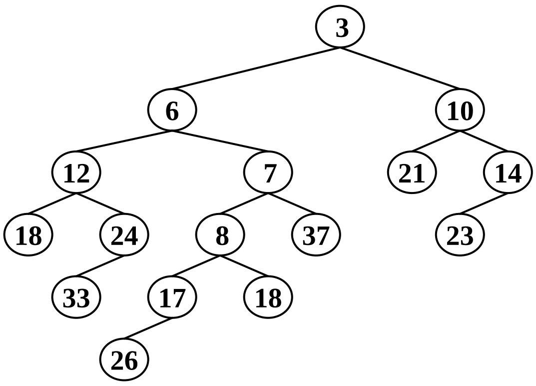
```
void OptMatrix( const long r[ ], int N, TwoDimArray M )
{
    int  i, k, Left, Right;
    long  ThisM;

    for( Left = 1; Left <= N; Left++ )  M[ Left ][ Left ] = 0;
    for( k = 1; k < N; k++ )
        for( Left = 1; Left <= N - k; Left++ ) {
            Right = Left + k;
            M[ Left ][ Right ] = Infinity;
            for( L = Left; L < Right; L++ ) {
                ①________________________________;
                if ( ThisM < M[ Left ][ Right ] )
                    ②________________________________;
            }
        }
}
```

III. Please write or draw your answers for the following problems on the answer sheet. (40 points)

(1) Please draw the results of inserting { 6, 7, 8, 9, 10, 1, 2, 3, 4, 5 } into
   a. (6 points)an initially empty AVL tree; and
   b. (7 points)an initially empty splay tree.
(Tip: Drawing the trees step by step might help you getting partial credits.)

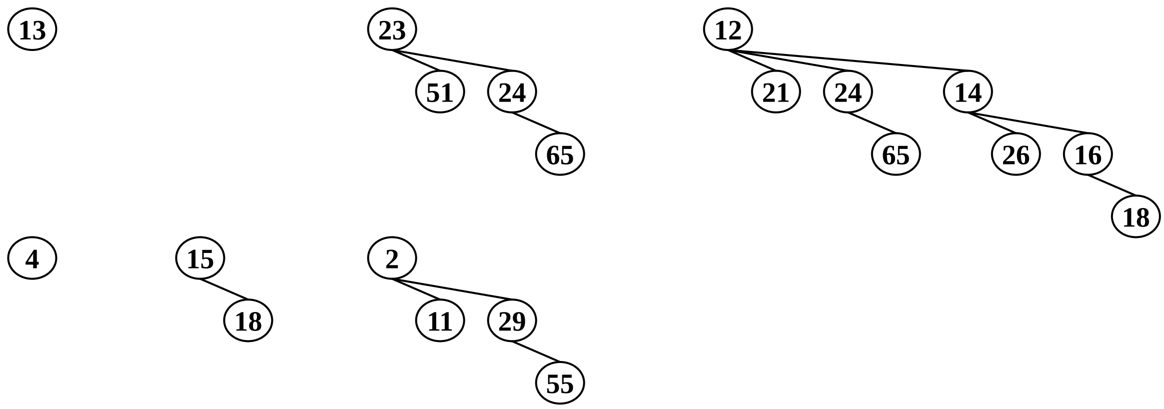
(2) Please draw the result of deleting the minimum number from the given leftist heap. (5 points)



(3) Please draw the result of merging the given two binomial queues. (5 points)

H1:

H2:



(4) Given a segment of text **this_is_a_strange_string**, how many bits are required to store the string as 1-byte characters? How many bits are needed to store its Huffman code? Please draw the Huffman tree with the following assumptions:
   a. All the characters are initially stored in an array in the alphabetical order; and
   b. All the left branches are coded with 0's and all the right branches are coded with 1's.
   (10 points)

(5) Please show the optimal binary search tree for the following words, where the frequency of occurrence is in parentheses: ***an*** (0.10), ***be*** (0.20), ***can*** (0.30), ***if*** (0.25), ***or*** (0.15). (7 points)

IV. Given the adjacency matrix G of a weighted graph, with G[k][k] presumed to be 0. Please write an algorithm with time complexity $O(N^3)$ for printing the shortest paths between any two vertices I, and j for all $0 \le I < j < N$. That is, your output must have the format as the following: (15 points)

```
0 -> … -> 1
… …
0 -> … -> N–1
1-> … -> 2
… …
1 -> … -> N–1
… …
N–2 -> … -> N–1
```

Answer Sheet

Part I						
(1)	(2)		(3)		(4)	(5)
(6)	(7)		(8)		(9)	(10)
(11)						

Part II	
(1)	(2)

Part III
(1)

(2)

(3)

(4)

(5)

Part IV