浙江大学 2009－2010 学年冬季学期

《高级数据结构与算法分析》课程期末考试试卷

课程号：____________，开课学院：软件学院、计算机学院、竺可桢学院

考试试卷：√A 卷、B 卷（请在选定项上打√）

考试形式：√闭、开卷（请在选定项上打√），允许带___无____入场

考试日期：2010 年 1 月 22 日，考试时间：120 分钟

诚信考试，沉着应考，杜绝违纪。

考生姓名：________________学号：____________________所属院系：_______________________

题序	一	二	三	四	总 分
得分					
评卷人					

Answer Sheet

Part I

1. d	2. a	3. c	4. d	5. a
6. c	7. b	8. c	9. a	10. b

Part II

1.

① this node has M+1 keys

② this node is the root

2.

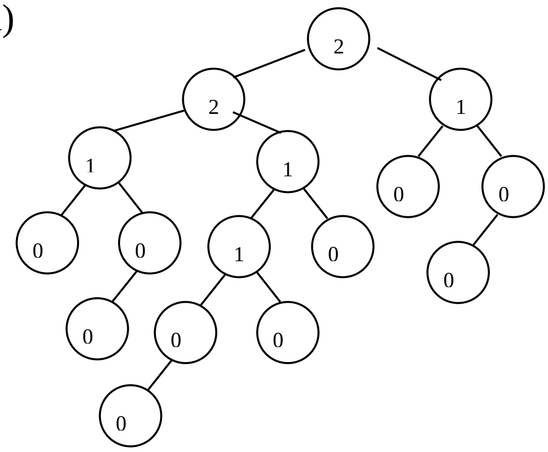
① H1->TheTrees[i] = H2->TheTrees[i] = NULL

② H1->TheTrees[i] = NULL

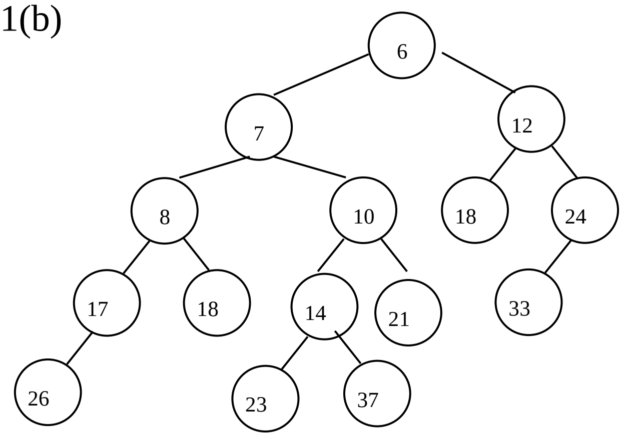
③ Carry = CombineTrees (T1, T2)

Part III

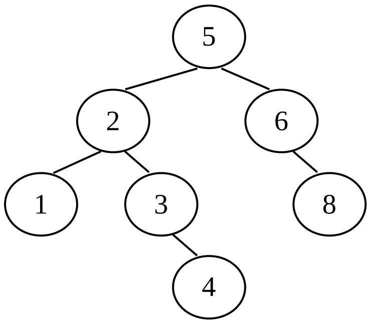
1(a)



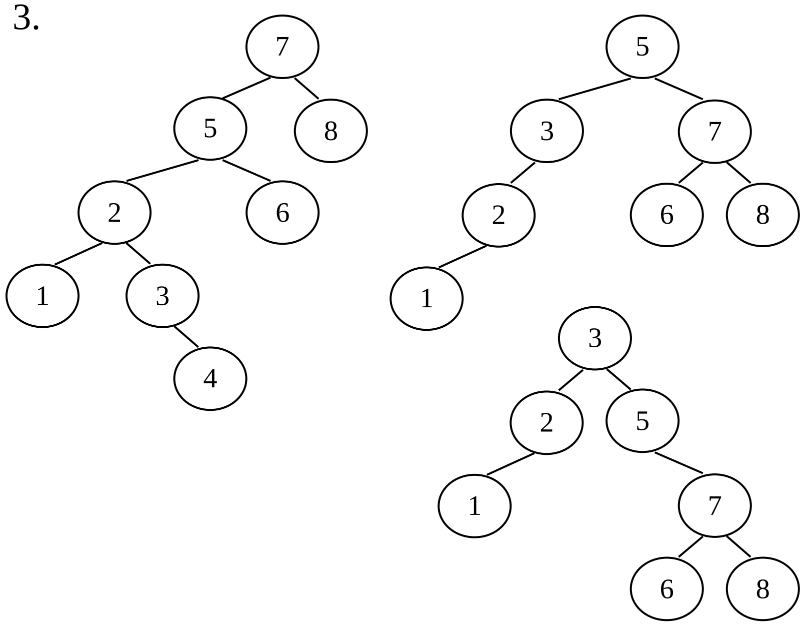
1(b)



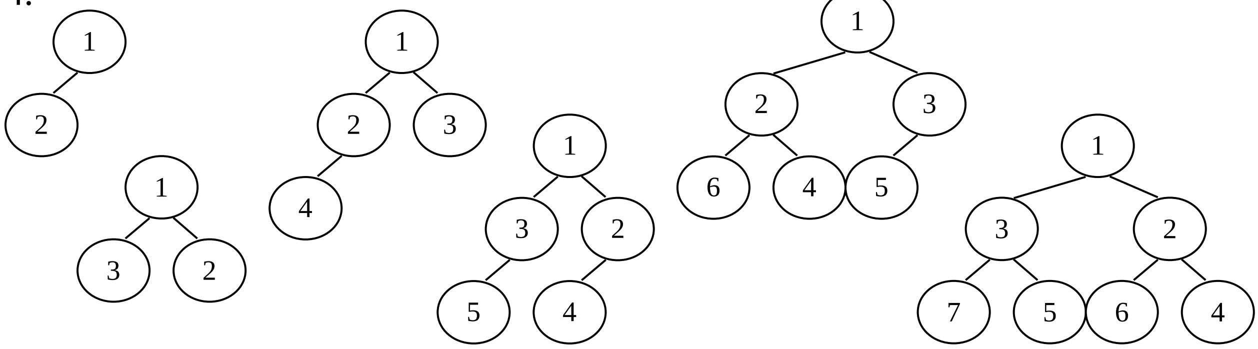
2.



3.



4.



5.

Operation / Heap type	Find Min	DeleteMin	Merge
Binary heap	O(1)	O(logN)	Θ(N)
Binomial heap	O(1) or O(logN)	O(logN)	O(logN)
Leftist heap	O(1)	O(logN)	O(logN)
Skew heap	O(1)	O(N)	O(N)

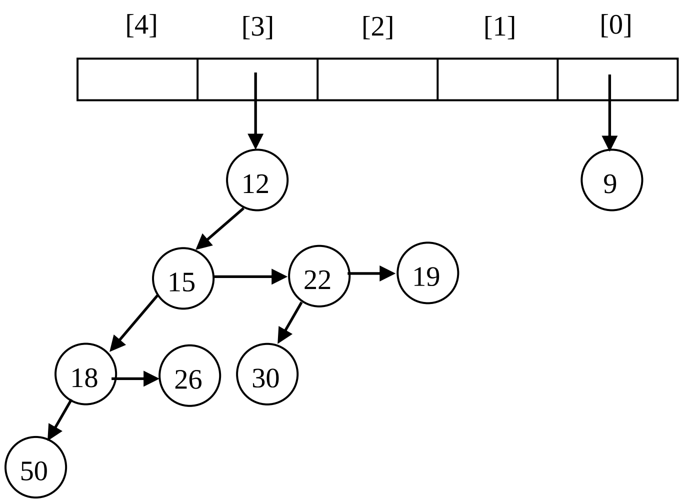
6(a).

```
Struct list_node {
   Int Doc_no;
   Int Line_no;
   Struct list_node *next;
};
Struct inverted_file_index {
   Int No;
   Char *term;
   Int times;
   Struct list_node * list;
} IFI[HASH_TABLE_SIZE];
```

6(b)

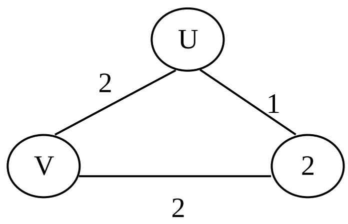
No	Term	Times; Documents,line
1	`trie`	<3; (1,1) ,(2,1), (3,2) >
2	`index`	<1; (1,1)>
3	`structure`	<1; (1,2)>
4	`useful`	<1; (1,2)>
5	`when`	<1; (1,3)>
6	`key`	<2; (1,3),(4,2)>
7	`vary`	<1; (1,3)>
8	`length`	<1; (1,4)>
9	`called`	<1; (2,1)>
10	`prefix`	<1; (2,2)>
11	`tree`	<3; (2,2),(3,4),(4,1)>
12	`you`	<1; (3,1)>
13	`suppose`	<1; (3,1)>
14	`introduce`	<1; (3,2)>
15	`compare`	<1; (3,3)>
16	`ordinary`	<1; (3,3)>
17	`binary`	<2; (3,4),(4,1)>
18	`search`	<2; (3,4),(4,1)>
19	`used`	<1; (4,2)>
20	`find`	<1; (4,2)>

7.



8.
No. Local optimal is not the global optimal.

Example:



Part IV

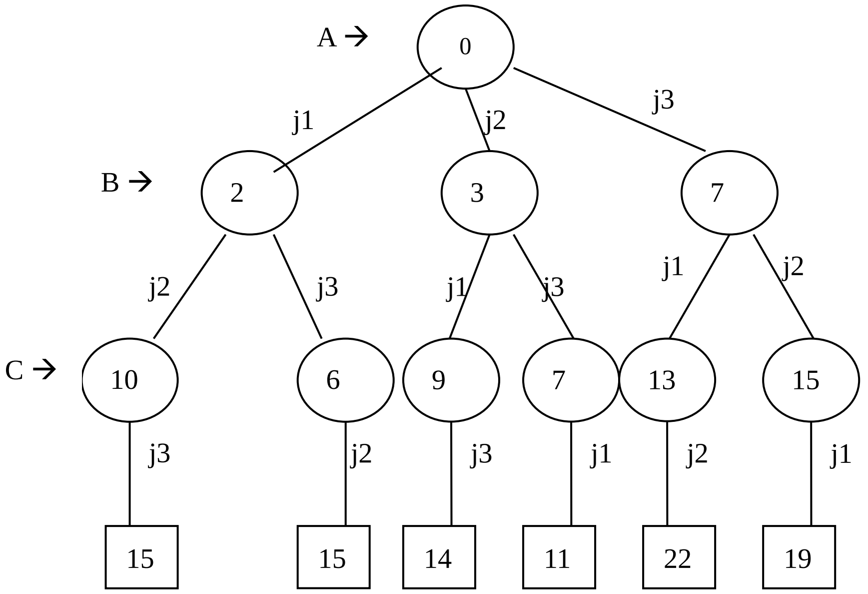
(1) The greedy strategy:

- find the minimum cost(i,j) in the matrix, assign job j to person i;
- delete column j and row i, and find the minimum cost(i,j) in the new matrix…

the solution is A-J1, B-J3, C-J2, cost=2+4+9=15

(2) The complete the decision tree as following, and the optimal solution is 11: A-J2, B-J3, C-J1.

(3) The pruning strategy: when the known feasible solution value greater than the cost of current node, prune the node. The pruned nodes during the search are: node(13) and node(15).



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 (the answer for each blank is unique). (2 points each)

1. For an AVL tree, __________ is NOT correct.
   a. A complete binary search tree must be an AVL tree
   b. In an AVL tree, the height of the left and right subtrees can differ by at most 1
   c. The left and right subtrees of an AVL tree are also AVL trees
   d. If the height of an empty tree is defined to be -1, an AVL tree of height 3 must contain exactly 7 nodes

2. Any M consecutive tree operations (such as insert, delete, find) starting from an empty splay tree of N nodes take ____________ time.
   a. at most O(M log N)
   b. at least O(M log N)
   c. at least O(log M × log N)
   d. at most O(N log M)

3. Which of the following statements concerning the B- and B+ trees is NOT true?
   a. Both B- and B+ trees are balanced trees
   b. Both B- and B+ trees can be used as index structures
   c. Both B- and B+ trees can be efficiently used for sequential search
   d. Both B- and B+ trees can be efficiently used for random search

4. A B+ tree of order 3 with 10 leaves has at most_____nonleaf nodes.
   a. 3   b. 4   c. 7   d. 8

5. Given two binomial queues H1 and H2, with thirteen and three nodes respectively. Let H3 be the binomial queue formed by merging H1 and H2. How many trees are there in H3?
   a. 1   b. 3   c. 13   d. 16

6. Among the following problems, ______ is NOT NP-complete.
   a. Vertex Cover
   b. Undirected Hamilton Circuit
   c. Undirected Euler Circuit
   d. Clique Cover

7. To solve a problem with input size N by divide and conquer algorithm, among the following methods, _____is the worst.
   a. divide into 3 sub-problems of equal complexity N/2 and conquer in O(NlogN)
   b. divide into 7 sub-problems of equal complexity N/2 and conquer in $O(N^2)$
   c. divide into 9 sub-problems of equal complexity N/3 and conquer in $O(N^2)$
   d. divide into 4 sub-problems of equal complexity N/3 and conquer in $O(N^2)$

8. The problem of "4 queens" is to place 4 queens on a 4×4 chessboard such that no two queens attack. A configuration is described by (x_1,x_2,x_3,x_4) where i is the row index of a queen and x_i is the column index where the ith row queen is placed. Which of the following statements are/is correct?

(1) There is no solution in the form $(1,x_2,x_3,x_4)$;

(2) The solution in the form $(2,x_2,x_3,x_4)$ is unique;

(3) There are only two solutions for this problem;

(4) The solution in the form $(4,x_2,x_3,x_4)$ is unique.

a. (1) b. (1) and (2) c. (1) ,(2) and (3) d. (2),(3) and (4)

9. Given 4 matrices and their sizes: M1(5×10), M2(10×4), M3(4×6), and M4(6×10). What is the optimal order of multiplications to obtain (M1×M2×M3×M4) with minimum number of computations?

a. (((M1×M2)×M3)×M4) b. ((M1×M2)×(M3×M4))

c. ((M1×(M2×M3))×M4) c. (M1×(M2×(M3×M4)))

10. In the all-pairs shortest path algorithm, a 2-dimensional array Path[][] is used to store the path. Assigning k to Path[i][j] means that vertex k is in the current path from i to j. Given the resulting Path matrix after applying the algorithm, and assume that a path exists between 1 and 4. What is the shortest path from 1 to 4?

	1	2	3	4	5
1			5	5	2
2			5	5	
3	5	5			
4	5	5			3
5	2			3	

a. 1->2->3->4 b. 1->2->5->3->4 c. 1->5->4 d. none of the above

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

1. The function is to insert element X into a B+ tree T of order M. (6 points)

```
Btree  Insert ( ElementType X,  Btree T )
{
    Search from root to leaf for X and find the proper leaf node;
    Insert X into the leaf node;
    while (  ① ____________________________  ) {
        split it into 2 nodes with ⌈(M+1)/2⌉ and ⌊(M+1)/2 ⌋ keys, respectively;
        if (  ② ____________________________  )
            create a new root with two children;
        check its parent;
    }
}
```

2. The function **Merge(BinQueue H1, BinQueue H2)** is to merge two binomial queues H1 and H2, with **CombineTrees(BinTree T1, BinTree T2)** defined as to merge two binomial trees T1 and T2 of the same size. (9 points)

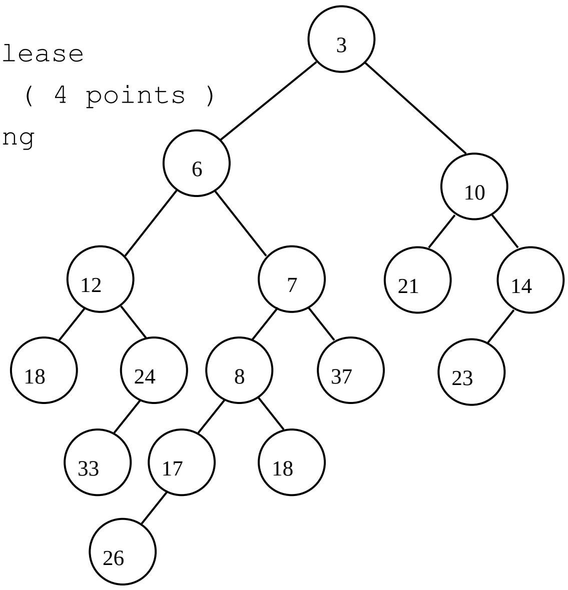
```
typedef struct BinNode *Position;
typedef struct Collection *BinQueue;
typedef struct BinNode *BinTree;
struct BinNode
{
    ElementType    Element;
    Position       LeftChild, NextSibling;
} ;
struct Collection
{
    int     CurrentSize;  /* total number of nodes */
    BinTree TheTrees[ MaxTrees ];
} ;

BinQueue Merge( BinQueue H1, BinQueue H2 )
{   BinTree T1, T2, Carry = NULL;
    int i, j;

    H1->CurrentSize += H2-> CurrentSize;
    for ( i=0, j=1; j<= H1->CurrentSize; i++, j*=2 ) {
       T1 = H1->TheTrees[i]; T2 = H2->TheTrees[i];
       switch ( 4*!!Carry + 2*!!T2 + !!T1 ) {
        case 0:
        case 1: break;
        case 2: H1->TheTrees[i] = T2; H2->TheTrees[i] = NULL; break;
        case 4: H1->TheTrees[i] = Carry; Carry = NULL; break;
        case 3: Carry = CombineTrees ( T1, T2 );
                ①__________________ ; break;
        case 5: Carry = CombineTrees ( T1, Carry );
                ② __________________; break;
        case 6: Carry = CombineTrees ( T2, Carry );
                H2->TheTrees[i] = NULL; break;
        case 7: H1->TheTrees[i] = Carry;
                ③_______________;
                H2->TheTrees[i] = NULL; break;
       } /* end switch */
    } /* end for-loop */
    return H1;
}
```

III. Please write or draw your answers for the following problems on the answer sheet. (50 points)

1. A leftist heap is given in the figure. Please
(a) mark the NPL values for all the nodes; (4 points)
(b) show the resulting heap after performing a DeleteMin. (5 points)



2. Please show the result of inserting 1, 2, 5, 3, 6, 8, 4 into an initially empty AVL tree. (6 points)

3. Please show the result of inserting 7 into the tree obtained from the above problem, and then the result of deleting 4, with ***splay*** rotations. (5 points)

4. Please show the result of inserting keys from 1 to 7 in increasing order into an initially empty skew heap. (7 points)

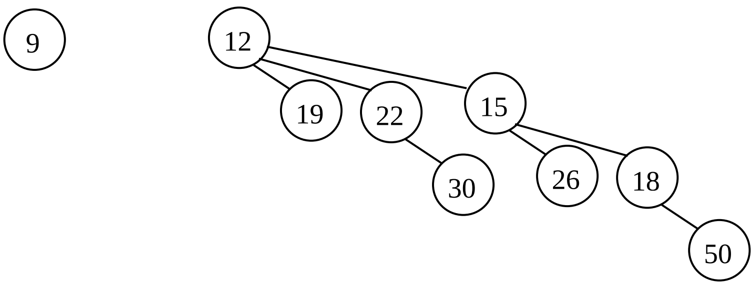
5. Please fill in the following table with the worst-case running time of performing the listed operations on a given heap with N nodes. (6 points)

Operation Heap type	Find Min	DeleteMin	Merge
Binary heap			
Binomial heap			
Leftist heap			
Skew heap			

6. Given a table listing the text in 4 documents to be indexed.
(a) Please design the data structure for representing the inverted file index. The structure is supposed to be convenient for searching the keywords *and* for easily printing the lines which contain the words. (3 points)
(b) Please list the inverted file index table for the given 4 documents. It is assumed that the stop words such as ***a, an, is, are, the, to, and, also, it, with*** have been eliminated from the original documents and word stemming has been performed. (5 points)

Doc	**Text**
1	**A trie is an index structure, it is useful when the keys vary in length.**
2	**A trie is also called a prefix tree.**
3	**you are supposed to introduce the tries and compare with ordinary binary search trees.**
4	**binary search trees are used to find keys.**

7. As we know, a binomial queue can be represented by an array of binomial trees, which are represented by linked tree structures. The following figure shows a binomial queue. Please draw the illustration of the array representation of this queue. (5 points)



8. Given as the following an algorithm for finding the shortest path between vertices U and V in a weighted undirected graph:

Step 1: Start from u = U and collect U into the current path;

Step 2: Be greedy when we choose the next vertex – that is, collect the closest vertex v to u into the path and then define u = v;

Step 3: Repeat Step 2 until u == V.

Is this algorithm correct? If your answer is "Yes", please prove it; else please give a counter example and explain your answer. (4 points)

IV. Job Allocation Problem

Job Allocation Problem is to assign N jobs to M (≥N) persons, with the cost of assigning the i-th job to the j-th person being a positive COST(i,j). We are supposed to find a way to minimize the total cost. For example, the following matrix gives the costs of 3 jobs (denoted by j1, j2, and j3) allocated to 3 persons (denoted by A, B, and C). If the allocation is A-j1, B-j2, C-j3, then the total cost is 15.
Note: Each person can do no more than 1 job.

	j1	j2	j3
A	2	3	7
B	6	8	4
C	4	9	5

(1) Please describe a greedy strategy to solve this problem, and show the result of applying your strategy to this example. (4 points)
(2) An optimal solution can be found by backtracking. A decision tree for this example can be constructed by making the non-leaf nodes represent the person, and the edges represent the possible jobs assigned. The value stored at each node is the current known cost. Please complete the given partial decision tree and find the optimal solution to this example. (8 points)
(3) Describe a pruning strategy during the backtracking search on the decision tree, and mark (X) at the pruned nodes. (3 points)