浙江大学 2013－2014 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： 21120490 ，开课学院：软件学院、计算机学院

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： 2014 年 1 月 19 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 (20)				
1.	2.	3.	4.	5.
6.	7.	8.	9.	10.

Part II (18)

1. ①________________ ②________________ ③________________	2. ①________________ ②________________ ③________________

Part III (47)

1. After deleting 0	1. After deleting 9

2. AVL insertion for 16

2. AVL insertion for 3

2. Splay insertion for 16

2. Splay insertion for 3

3.

4,	5.
6.	
Part IV (20)	

第 4 页 共 9 页

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. Which one of the following is FALSE about Tries trees and AVL trees: they are both _______.

a. fit for random searches b. fit for range searches

c. balanced binary trees d. fit for sequential searches

2. For a splay tree, ______ is correct.

a. Any single operation might take O(N) time, thus a splay tree has an O(N) amortized cost per operation

b. Splay trees maintain balance information of the tree

c. The relation of splay trees to AVL trees is analogous to the relation between binomial queues and priority queues

d. After a node is accessed, it is pushed to the root

3. Which of the following statements concerning a B+ tree of order M is FALSE?

a. The root has at most M children

b. All leaves are at the same depth

c. All nonleaf nodes have between $\lceil M/2 \rceil$ and M children

d. Keys in any node are ordered

4. Which one of the following is FALSE about leftist heaps?

a. A leftist heap with N nodes on the right path must have at least 2^N-1 nodes

b. *DecreaseKey* can always be efficiently supported by leftist heaps in O(logN) where N is the total number of nodes

c. A perfectly balanced tree forms if keys 1 to 2^N-1 are inserted in order into an initially empty leftist heap

d. The worst case time bound of merging two leftist heaps of size N is O(logN)

5. After insert 1,2,3,…, 15 in order into an initially empty binomial queue, in which tree of this binomial queue that the number 11 will be?

a. B_0 b. B_1 c. B_2 d. B_3

6. Given M tourist groups (旅游团) with A_i members in the ith group for i = 1, …, M. Suppose that we only have one flight with N ($\le \sum A_i$) seats. How can we select the groups to best fit all the seats? This problem can be best solved by

a. dynamic programming b. divide and conquer

c. greedy method d. backtracking

7. Which one of the following algorithms uses the divide and conquer technique?

a. Merge leftist heaps b. Eight queens problem

c. Quick sort d. Dijkstra’s algorithm for the shortest path

8. To pack the given 7 items of sizes 0.2, 0.3, 0.1, 0.7, 0.5, 0.8, 0.4 into the bins with unit capacity by the *off-line* algorithms, the optimal solution can be obtained by ________.

a. first-fit and best-fit   b. next-fit and best-fit

c. best-fit only   d. all the three fitting methods

9. Which one of the following statements is TRUE?

a. Suppose there is a polynomial-time reduction from Problem A to Problem B, we can draw a conclusion that, if A is NP-complete, then B is NP-complete

b. Undecidable problems are still undecidable, even if nondeterminism is allowed

c. Halting problem is NP-complete

d. Undirected Euler circuit is NP-complete

10. Which one of the following statements about amortized time is FALSE?

a. The amortized time to merge two skew heaps of size N is O(logN)

b. Average bounds are weaker than amortized bounds

c. A binomial queue of N elements can be built by N successive insertions in O(N) time, although the worst-case time for each insertion is O(logN)

d. When analyzing amortized bounds, a good potential function usually assumes its maximum at the start of the sequence

II. Given the function descriptions of the following two (pseudo-code) programs, please fill in the blank lines. (18 points)

1. The function is to merge two binomial queues H1 and H2. (9 points)

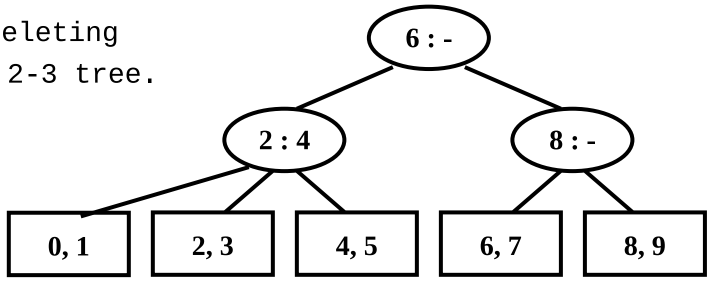
```
BinQueue  Merge( BinQueue H1, BinQueue H2 )
{   BinTree T1, T2, Carry = NULL;
    int i, j;
    H1->CurrentSize += H2-> CurrentSize;
    for ( i=0, j=1; j<= H1->CurrentSize; i++, j*=2 ) {
        T1 = H1->TheTrees[i]; T2 = H2->TheTrees[i];
        switch( 4*!!Carry + 2*!!T2 + !!T1 ) {
         case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

2. The function is to find the shortest paths for all pairs of vertices. The arrays A contains the adjacency matrix, D contains the shortest path length, and Path can be used to compute the actual path. (9 points)

```
void AllPairs( TwoDimArray A, TwoDimArray D, TwoDimArray Path, int N )
{   int  i, j, k;
    for ( i = 0; i < N; i++ )  /* Initialize D and Path*/
        for( j = 0; j < N; j++ ) {
            D[ i ][ j ] = A[ i ][ j ];
            Path[ i ][ j ] = NotAVertex;
        }
    for( k = 0; k < N; k++ )
        for( i = 0; i < N; i++ )
            for( j = 0; j < N; j++ )
                if(①____________________________________) {
                    D[ i ][ j ] = ②________________________;
                    Path[ i ][ j ] = ③_____________________;
                }
}
```

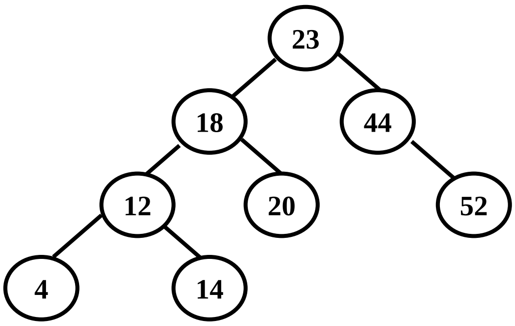
III. Please write or draw your answers for the following problems on the answer sheet. (42 points)

1. Please draw the results of deleting 0 and then 9 from the given 2-3 tree. (6 points)



2-3 tree for problem 1

2. Please draw the results of inserting 16 and then 3 into the given tree as an AVL tree, and as a splay tree. (8 points)



Tree for problem 2

3. One way of defining the distance of any given pair of documents is to calculate the inner product of the two vectors containing the word frequencies for all words in two documents. That is, Denote the frequency vector for document D_i by $\vec{F}(D_i)=(F_i(w_1),\cdots,F_i(w_n))^T$, where $F_i(w_j)$ is the frequency of word w_j in document D_i, then the metric is given by

$$(\vec{F}(D_1),\vec{F}(D_2))=\sum_{i=1}^{n}F_1(w_i)\cdot F_2(w_i)\,.$$

Given two documents D_1 and D_2, please *briefly* describe how to construct the inverted file index and how to calculate the document distance fast. Analyze the complexity of your method. (10 points)

4. Given 7 words and the probabilities of searching them.

WORD	break	case	char	do	return	switch	void
PROBABILITY	0.22	0.18	0.20	0.05	0.25	0.02	0.08

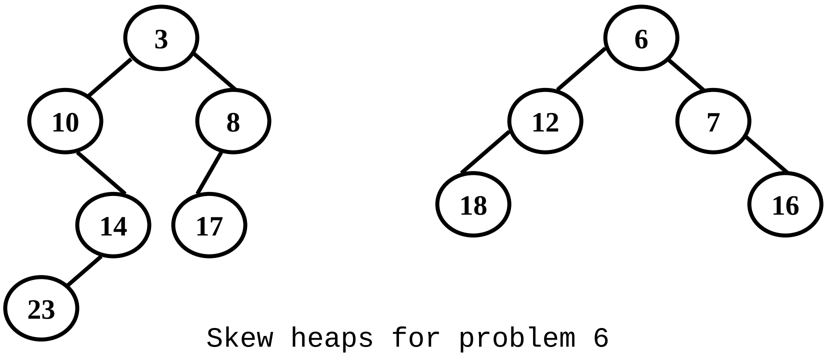
The following table is for constructing the optimal binary search tree.

break.. break		case..case		char.. char		do..do		return..return		switch..switch		void.. void	
0.22	break	0.18	case	0.20	char	0.05	do	0.25	return	0.02	switch	0.08	void
break.. case		case.. char		char..do		do.. return		return..switch		switch.. void			
0.58	break	0.56	char	0.30	char	0.35	return	0.29	return	0.12	void		
break.. char		case..do		char.. return		do.. switch		return.. void					
1.02	case	0.66	char	0.80	return	0.39	return	0.47	return				
break..do		case.. return		char.. switch		do.. void							
1.17	case	1.21	char	0.84	return	0.57	return						
break.. return		case.. switch		char.. void									
1.83	char	1.27	char	1.02	return								
break.. switch		case.. void											
1.89	char	1.53	char										
break.. void													
2.15	char												

Please draw the optimal binary search tree for the words: case, char, do, return, and switch. (6 points)

5. In a turnpike reconstruction problem, the distance set is given as {1, 2, 3, 3, 4, 6, 7, 7, **X**, 10}, and the point set is {0, 3, **Y**, 9, 10}. Please find the values of **X** and **Y**. (6 points)

6. Please draw the result of merging two given skew heaps. (6 points)



Skew heaps for problem 6

IV. Monkeys and Bananas (20 points)

Monkeys love bananas. Suppose that there were N monkeys, each found a pile of B[i] bananas where i=1, …, N. They wanted to merge all the bananas into one big pile by merging two piles at a time. If the energy used to merge any two piles is proportional to (成正比) the total size of the two piles, how would you suggest them to get the work done with minimum amount of energy?

(1) Please describe the algorithm (12 points) and

(2) analyze the time complexity (3 points).

(3) If all the banana piles were placed in a line, and they could only merge two adjacent piles at a time, then how would you suggest them to do? Please briefly describe your idea and analyze the time complexity as well. (5 points)