Department of Computer Science and Engineering
National Sun Yat-sen University
Data Structures - Final Exam., Jan. 11, 2016

1. Multiple choices (There may be zero or more correct answers. If there is no correct answer, you should write down "None".) (20%)

   (a) Which statement(s) is correct for a tree? (A) If a node is to be deleted, it must be a leaf node. (B) In a binary search tree, the height difference of the left and right subtrees of every node is always at most 1. (C) In a binary tree, a newly inserted node is always a leaf node. (D) If each node of a tree has at most two children, then it is a binary tree.

   (b) Which statement(s) is correct for a red-black tree? (A) The color of the root is red. (B) All leaf nodes are on the same level. (C) In the path from the root to each external node, there are no two consecutive red nodes. (D) When a new node is inserted into the tree, its color is set to red.

   (c) Which statement(s) is correct for the hashing technique? (A) Hashing is a fast searching method. (B) If the chaining method is used, then the data elements with the same position index are chained by the linked lists. (C) If the linear probing method is used, then an element in the hash table can be deleted directly. (D) Linear probing is one of the methods for resolving hash collisions.

   (d) Which statement(s) is correct for an AVL tree? (A) If we traverse the tree nodes with the preorder sequence, then the sequence is sorted. (B) LL rebalancing rotation is symmetric to RR rebalancing rotation. (C) In the LR rebalancing rotation, one of the tree nodes is moved up 2 levels. (D) The level difference of every two nodes is at most 1.

   (e) Which statement(s) is correct for a B+ tree? (A) All data elements are stored in the external nodes. (B) All leaves are on the same level. (C) The data nodes are linked together. (D) If a new data element is inserted into the tree, then it is always inserted into the external nodes.

2. (a) Give the definition of a complete binary tree. (4%)

   (b) Suppose that a complete binary tree is stored in an array a[ ], with the root stored in a[1]. Please give the parent of a[i] and the children of a[i]. (6%)

3. [3 2 8 5 7 1 4 6] is a permutation of [1 2 3 4 5 6 7 8]. Please give the two nontrivial permutation cycles in [3 2 8 5 7 1 4 6]. (6%)

4. Huffman algorithm is a variable-length encoding method, in which a Huffman tree is constructed. Suppose seven symbols A, B, C, D, E, F, G with frequencies 10, 7,

6, 3, 15, 1, 6, respectively, are given. You have to obey the following rules for constructing the Huffman tree:

(b) The label of each leaf node is a single symbol. The label of each internal node is the concatenation of the labels of its left child and right child.

(c) When two nodes are merged, always set the node of label with less lexical order as the left child, and the other as the right child.

(d) If more than two nodes have the same least frequency, then you have to merge the two that have the least and the second least label in lexical order.

Please draw the full Huffman tree. (10%)

5. Suppose that the division method (mod 13) is used in the hashing function. The linear probing method is applied when a collision occurs. Given input numbers: 5, 13, 17, 4, 19, 11, 26, 25, please show the hash table after all numbers have been inserted into the table according to the input order. (10%)

6. Please give the three important properties a B-tree of order m, including the number of children of the root, the number of children of each node (other than the root and the external node), the positions of the leaf nodes. (10%)

7. We are given a set $T=\{t_1, t_2, t_3, \ldots, t_n\}$ of numbers, and a number M. The perfect pair problem is to find a pair of numbers t_i and t_j, $i \neq j$, in T such that $t_i+t_j = M$. For example, suppose T={10, 15, 3, 13, 8, 11, 4, 5}, and M=17. Then, the answer is 13 and 4. Please design an algorithm to solve the perfect pair problem and use the example to explain your algorithm. Note that your algorithm should be in O($n$log$n$) time. Analyze the time complexity. (10%)

8. Write a recursive C++ function to find the maximum value of all data stored in a binary tree. Note that every node stores one data element. (12%).

```
class TreeNode {
   int data;
   TreeNode *leftChild, *rightChild;;
};
int maximum(TreeNode *root)
// Return the maximum of the data stored in the binary tree pointed by “root”.
// Return -∞ if the binary tree is empty.
{
```

Please write the body of the function.

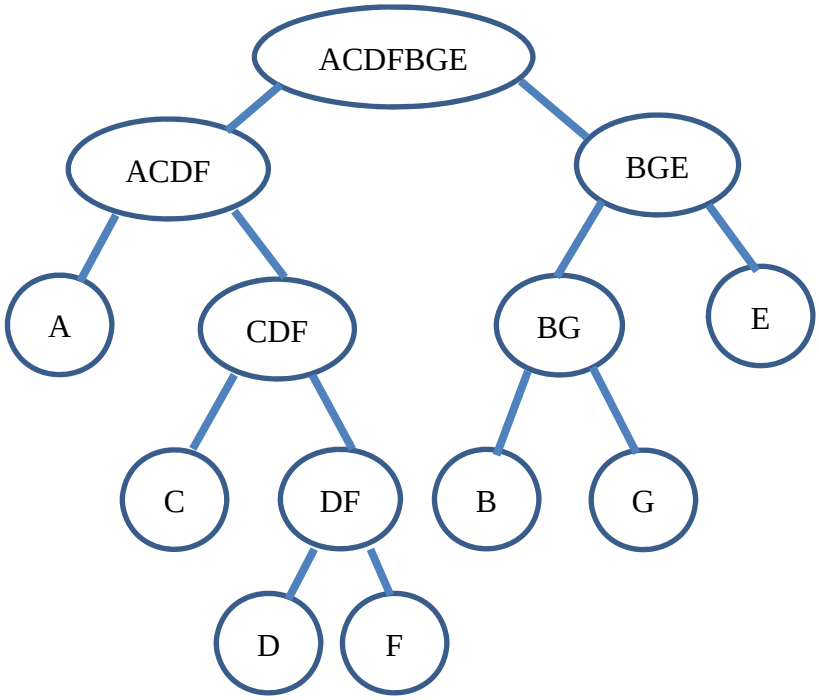
```
} // end of maximum( )
```

9. Write a C++ function to perform the insertion sort, where the sorting result is in nondecreasing order. (12%)

```
void insertion(int a[ ], int n)
/* a[ ]: the data array,    n: number of elements in a[ ] */
```

Answer:

1. C, CD, ABD, BC, ABCD.
2.
   (a) Complete Binary tree with depth k is that depth 1 ~ k-1 is full, and nodes in depth k must keep left.
   (b) Parent of a[i] is $\lfloor i/2 \rfloor$, children of a[i] are 2i, 2i+1.
3. (1, 3, 8, 6) and (4, 5, 7)
4.



5. hash table

0	1	2	3	4	5	6	7	8	9	10	11	12
13	26			17	5	4	19				11	25

6. B-tree
   (1) $2 \le$ # children of root $\le m$
   (2) All nodes other than the root node and external nodes:
   $\lceil m/2 \rceil \le$ # children $\le m$
   (3) All external nodes are at the same level.

7. Perfect pair problem : 給定一個集合，有 n 個數字，以及一個數字 M。從集合裡找出兩個數字，使得兩數相加等於 M。
   解法 1:
   第一步:先使用 merge sort 將資料由小到大排序。
   第二步:從最小的數字 t_i 開始枚舉 $M - t_i$，並從剩餘較大的數字中使用 binary search 找尋答案。
   以例子 T={10, 15, 3, 13, 8, 11, 4, 5}且 M=17 為例：

先將資料排序：T={3, 4, 5, 8, 10, 11, 13, 15}

1. 先挑最小的 3

binary search 範圍 : 4 ~ 15，正中間的數: 10

3+10 = 13 < 17 因此往右邊搜尋

binary search 範圍 : 11 ~ 15，正中間的數: 13

3+13 = 16 < 17 因此往右邊搜尋 … 依序做，發現找不到答案

2. 接著挑第二小的 4

binary search 範圍 : 5 ~ 15，正中間的數: 10

4+10 = 14 < 17，因此往右邊搜尋

binary search 範圍 : 11 ~ 15，正中間的數: 13

4+13 = 17，找到答案！

時間複雜度分析：

第一步 merge sort : O(nlogn)

第二步枚舉所有數字並將剩餘數字做 binary search：O(n)*O(logn) = O(nlogn)

Time complexity: O(nlogn)

解法 2:

第一步:先使用 merge sort 將資料由小到大排序。

第二步:先挑選最大與最小的數字檢查是否為答案。

若相加比 M 還大，表示最大的數字一定不是答案(因為拿最小的數字與它相加依然太大)，因此檢查最小與第二大的數字。

同理，若相加比 M 還小，表示最小的數字一定不是答案(因為拿最大的數字與它相加依然太小)，因此檢查第二小與最大的數字。

依序往內檢查直到找到答案。(Greedy method)

時間複雜度分析：

第一步 merge sort : O(nlogn)

第二步枚舉所有數字一次：O(n)

Time complexity: O(nlogn)

8.

```
class TreeNode {
public:
    int data;
    TreeNode *leftChild, *rightChild;;
};
int maximum( TreeNode *root )
{
    int leftM, rightM;
    if( root == 0 )
        return -∞;      // Return -∞ if the binary tree is empty.
    leftM = maximum( root->left );
    rightM = maximum( root->right );
```

```
    if ( leftM >= rightM )
        return root->data>leftM ? root->data : leftM;          // left subtree is larger
    else
        return root->data>rightM ? root->data : rightM;       // right subtree is larger
} // end of maximum( )
```

9.

```
void insertion(int a[ ], int n)
/* a[ ]: elements are stored in a[0], a[1], …, a[n-1];
   n: number of elements in a[ ] */
{
    int i,j;
    int temp;

    for (i=1; i<=n-1; i++) {
       temp=a[i];
       for (j=i-1; j>=0; j--) {
          if (temp < a[j])
             a[j+1]=a[j];      // move a[j] to the right
          else {
             a[j+1]=temp;      // insert the element to a[j+1]
             break;            // quit this iteration
          }    // end of else
       }    // end of for
    }    // end of for
}     // end of insertion( )
```