四川大学期末考试试题（闭卷）

（2017~2018 学年第 1 学期） B 卷

课程号：**311076040** 课程名称：**数据结构与算法** 任课教师：

适用专业年级：**软件工程 2016 级** 学号： 姓名：

考生承诺

我已认真阅读并知晓《四川大学考场规则》和《四川大学本科学生考试违纪作弊处分规定（修订）》，郑重承诺：

1、 已按要求将考试禁止携带的文具用品或与考试有关的物品放置在指定地点；

2、 不带手机进入考场；

3、 考试期间遵守以上两项规定，若有违规行为，同意按照有关条款接受处理。

考生签名：

题　　号	一(30%)		二(16%)	三(34%)		四(20%)
得　　分						
卷面总分		教师签名		阅卷时间		

注意事项： 1. 请务必将本人所在学院、姓名、学号、任课教师姓名等信息准确填写在试题纸和添卷纸上；

2. **请将答案全部填写在答题纸上；本试题纸上的答案一律不计分；**

3. 考试结束，请将试题纸、添卷纸和草稿纸一并交给监考老师。

评阅教师	得分

一、单项选择题（本大题共 15 小题，每小题 2 分，共 30 分）

提示： 在每小题列出的四个备选项中只有一个是符合题目要求的，请将其代码写在答题纸上。错选、多选或未选均无分。

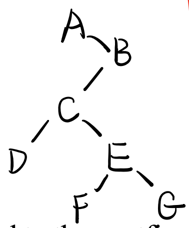
1. If the preorder of a binary tree is ABCDEFG, then the possible inorder is ( ).

（A） ABCDEFG

（B） CABDEFG

（C） DACEFBG

（D） ADCFEGB



2. An arithmetic expression a+b*（c+d/e）can be changed to the postfix expression ( ).

（A） ab+cde/*

（B） abcde/*++

（C） abcde/+*+

（D） abcde*/++

3. What is the worst case time complexity for search in a general tree? ( )

（A） log(n)

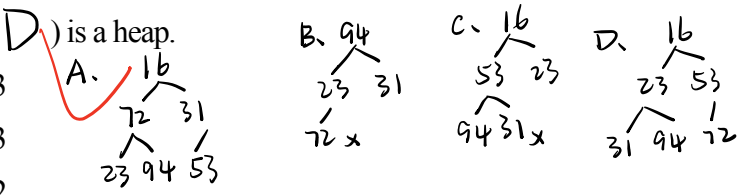
（B） n

（C） nlogn

（D） n^2

（E） None of above

4. In the following sequence, (D) is a heap.



（A） 16, 72, 31, 23, 94, 53

（B） 94, 23, 31, 72, 16, 53

（C） 16, 53, 23, 94, 31, 72

（D） 16, 23, 53, 31, 94, 72

5. How many linked lists are used to represent a graph with n nodes and m edges, when using an adjacency list representation.(D)

（A） m + n

（B） m

（C） m * n

（D） n

6. An algorithm must be or do all of the following EXCEPT (C)

（A） correct

（B） composed of concrete steps

（C） ambiguous

（D） composed of a finite number of steps

7. The best data structure to check whether an arithmetic expression has balanced parentheses is (B)

（A） queue

（B） stack

（C） tree

（D） linked list

8. Which structure is convenient for dynamic inserting and deleting (B)

（A） array

（B） link list

（C） stack

（D） queue

9. If the sequence {11, 12, 13, 7, 8, 9,23, 4, 5} is the middle result after one pass, then the sort method used is (B).

（A） Bubble sort

（B） Insertion sort

（C） Selection sort

（D） Two-way Mergesort

10.If the MaxSize of a Circular Queue is n and there is always a space not use, front points to the

previous of the front element in the queue, and rear points to the rear element in the queue. The number of items in the Queue can be expressed by (A).

(A) (rear -front + n) % n

（B） rear-front+1

（C） rear-front-1

（D） rear-front

11.If the height of a Complete Binary Tree is n, then the number of node is at most (C)

（A） 2n

（B） n

$2^0+\cdots+2^{n-1}=\frac{1-2^n}{1-2}=2^n-1$

(C) 2^n – 1

（D） 2^(n-1) – 1

12.If a Huffman tree has 199 nodes, the Huffman tree has (B) leaf nodes.

$n_0=n_2+1$

$199=n_0+n_2=2n_2+1 \Rightarrow n_2=99$

$n_0=100$

（A） 99

（B） 100

（C） 101

（D） 102

13.Dijkstra's algorithm requires that vertices be visited in (C)

（A） Depth-first order.

（B） Breadth-first order.

（C） Order of distance from the source vertex.

（D） No particular order.

14.The 80/20 rule indicates that (A)

(A) 80% of the searches in typical databases are to 20% of the records.

(B) 80% of searches in typical databases are successful and 20% are not.

(C) 80% of records in typical databases are of value, 20% are not.

15.A sorting algorithm is stable if it (C)

（A） Works for all inputs

（B） Always sorts in the same amount of time (within a constant factor) for a given input size.

(C) Does not change the relative ordering of records with identical key values.

（D） none of the above

评阅教师	得分

二、名词解释题（本大题共 4 小题，每小题 4 分，共 16 分）。

提示：解释每小题所给名词的含义，若解释正确则给分，若解释错误则无分，若解释不准确或不全面，则酌情扣分。

1. ADT
2. MST
3. Full Binary Tree
4. Quick Sort

评阅教师	得分

三、应用题（本大题共 4 小题，1-2 每小题 8 分，3-4 每小题 9 分，共 34 分）

提示：有求解过程的要尽量给出解题步骤，只有最终答案会酌情扣分。

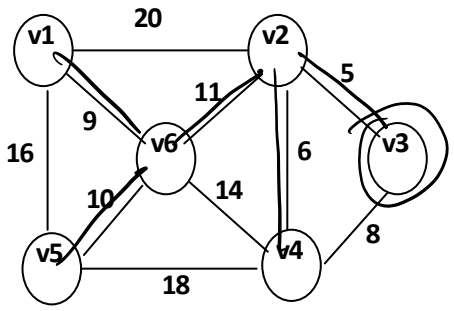
1. Assume that you have a seven-slot closed hash table (the slots are indexed 0 through 6).

   1) Show the final hash table if you use the hash function H(k) = k mod 7 and the simple linear probing $d_i = i$, on this list of numbers: 18, 50, 71, 25

   2) After inserting the above numbers, calculate the probability for each empty slot that it will be the next one filled.

   3) Detemine the SL(关键字比较次数) when searching 71 in the HT

2. Given the following undirected graph,



1) List the order of the edges which are added into MST when running Prim's MST algorithm. Starting at vertex 3.

2) Show the final MST.

3. You are given a series of records whose keys type is int, the records arrive in the following order:

   18 50 10 20 31 12 23 33,

   1) Show the process of constructing a B+ whose internal nodes can store up to 4 children and whose leaf nodes can store up to 5 records from inserting these records.

   2) Show the result of deleting the value 12 from the B+ tree of 1).

4. Assume that a sample alphabet has the following weights:

Letter	A	B	C	D	E	F	G	H	I
Frequency	3	5	9	15	20	22	36	39	50

(a)Build the Huffman coding tree and determine the codes for the letters.

(b)What is the average number of bits required by a character using the Huffman code for this alphabet?

评阅教师	得分

四、编程、设计及分析题（本大题共 2 小题，1 小题 8 分，2 小题 12 分，共 20 分）。

提示：每小题给出了一个程序设计要求，请按照要求写出源程序代码，如果源程序代码中出现语法错误或逻辑错误，则酌情扣分。

1. Write a BFS (Breadth First Search) function of a binary tree. (8 points)
2. Write a function to determine whether a tree is an AVL tree. (12 points)

左右两个子树高度差的绝对值不超过1的特殊二叉检索树

1. (1) H(18) = 4

H(50) = 1

H(71) = 1　　H(71)+d1 = 2

H(25) = 4　　H(25)+d1 = 5

final Hash Table

→	0	1	2	3	4	5	6
		50	71		18	25	

(2) 落入slot 0 = P = 1/7

落入slot 3 : P = 3/7

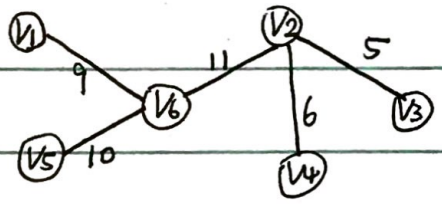
落入slot 6 = P = 3/7.

(3) probe sequence = 1、2

∴ SL = 2.

2. (1) V_3-V_2、V_2-V_4、V_2-V_6、V_6-V_1、V_5-V_6.

(2) MST =

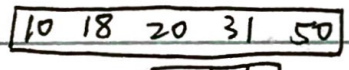


3. (1) 对 root : 关键码数 : 1 ∼ m 即 1 ∼ 3 Child数 : 2 ∼ 4

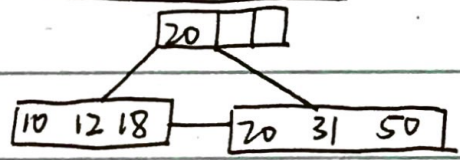
对 internal nodes : keys数 $\lceil \frac{m}{2} \rceil - 1 \sim m-1$: 1 ∼ 3　　Child数 : 2 ∼ 4

对 leaf nodes = keys数 $\lceil \frac{n}{2} \rceil \sim n$ 即 3 ∼ 5.

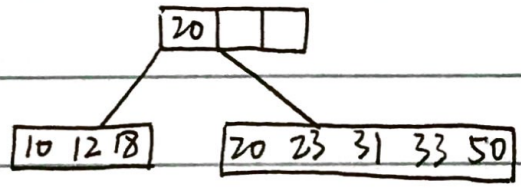
插入 18 ∼ 31 :



插入 12 :

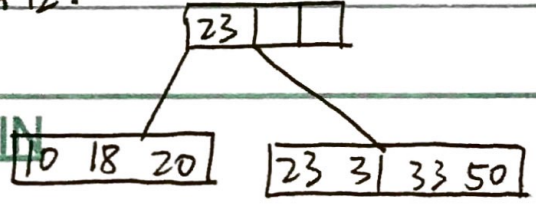


插入 23、33 :

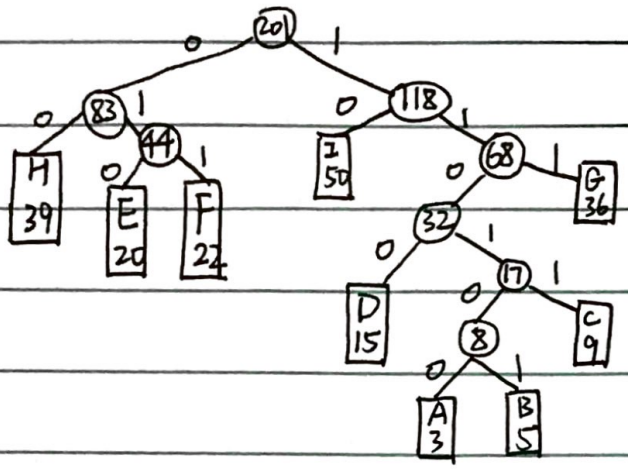


(最终 B+ 树)

(2) 删去 12 :



4.10) Huffman Coding Tree:



A: 110100 (6bit) D: 1100 (4bit) G: 111 (3bit)

B: 110101 (6bit) E: 010 (3bit) H: 00 (2bit)

C: 11011 (5bit) F: 011 (3bit) I: 10 (2bit)

(b) Average Number = $\dfrac{3\times6+5\times6+9\times5+15\times4+20\times3+22\times3+36\times3+39\times2+50\times2}{3+5+9+15+20+22+36+39+50}$

$= \dfrac{565}{199} = 2.84$

```
void BFS(BinNode<E>* root){
    BinNode<E> * Now;
    LQueue Q(BinNode<E>*);
    enqueue(root);
    cout<< root->getValue()<<endl;
    while(Q.length()!=0){
        Now = Q.dequeue();
        if(Now->leftChild()!==NULL) Q.enqueue(Now->leftChild()
        if(Now->RightChild()!=NULL) Q.enqueue(Now->rightChild()
        cout<<Now->getValue()<<endl;
    }
}
```

2. int heightTree(BinNode<E>* subroot){

if(subroot == NULL) return 0;

int leftH = heightTree(subroot->leftChild());

int rightH = heightTree(subroot->rightChild());

return (leftH>rightH)?(1+leftH):(1+rightH);

}

bool isBalance(BinNode<E>* subroot){

if(subroot == NULL) return true;

if(isBalance(subroot->leftChild())== true && isBalance(subroot->rightChild())

== true){ //先判断子树是否平衡

int leftH = heightTree(subroot->leftChild());

int rightH = heightTree(subroot->rightChild());

return ((-1<=leftH-rightH)&&(leftH-rightH<=1))?(true:false);

}

else return false; //子树不平衡时返回false

}

bool isBST(BinNode<E>* subroot){

if(subroot == NULL) return true;

if(isBST(subroot->leftChild())== true && isBST(subroot->rightChild())

== true){

subroot->isleaf() == true

if(isleaf(subroot)==true) return true; //只有1个子结点，是BST树

else if(subroot->leftChild()->getValue() < subroot->getValue()

&& subroot->rightChild()->getValue()>=subroot->getValue())

return true;

else return false;

}

else return false; //左右子树不为BST返回false

}

else if

注：还有两种情况

1）左子树为空

2）右子树为空

```
bool isAVL(BinNode<E>* subroot){
    if(subroot==NULL) return true;
    if(isAVL(subroot->leftChild())==true && isAVL(subroot->rightChild())==true){
        return (isBalance(subroot)==true && isBST(subroot)==true)?
            true:false;
    }
    else return false;
}
```