

# 考试科目名称 数据结构 (A 卷答案)

考试方式：开卷 闭卷 考试日期\_\_\_\_年\_\_月\_\_日 教师 陈珮珮

系（专业） 软件学院 年级 二年级(07 级) 班级\_\_\_\_\_

学号\_\_\_\_\_ 姓名\_\_\_\_\_ 成绩\_\_\_\_\_

题号	一	二	三	四	五	六	七	八	九	十
分数										

得分	
----	--

## 1. 算法分析题（10 分）

i: 1 2 3 4.....n-1

j: 2~n 3~n 4~n 5~n.....n~n

k: 1~j

2 + 3 + 4 + 5 + ..... + n

+ 3 + 4 + 5 + ..... + n

+ 4 + 5 + ..... + n

+ 5 + ..... + n

+ .....

+ n

= 2 \* 1 + 3 \* 2 + 4 \* 3 + 5 \* 4 + ..... + n \* (n-1)

= 1<sup>2</sup> + 1 + 2<sup>2</sup> + 2 + 3<sup>2</sup> + 3 + 4<sup>2</sup> + 4 + .... + (n-1)<sup>2</sup> + (n-1)

= 1<sup>2</sup> + 2<sup>2</sup> + 3<sup>2</sup> + 4<sup>2</sup> + ..... + (n-1)<sup>2</sup> + 1 + 2 + 3 + 4 + ... + (n-1)

= (n-1) n (2n-1)/6 + (n-1) n/2 = n (n-1) (n + 1)/3

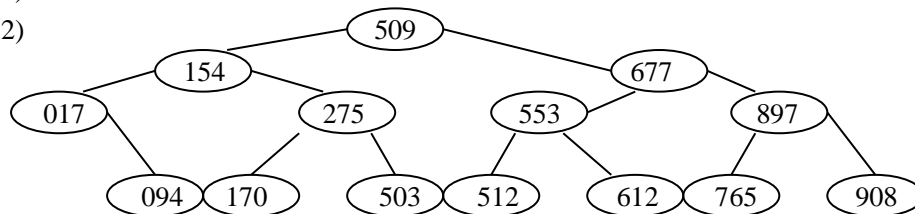
= O(n<sup>3</sup>)

得分	
----	--

## 2. （20 分，每题 5 分）

1) 2\*k-1 2<sup>k</sup>-1

2)



$$AVLsucc = (1 + 2 * 2 + 3 * 4 + 4 * 7) / 14 = 45 / 14$$

3). xsxxsxxsxxsxxssss

4). 216 72 310 = 100 + (8 \* 4 + 3) \* 6

得分

3. (10 分)

1	3	7	
1	2	11	
1	3	4	12
1	3	5	14
1	3	6	15

得分

4. (10 分)

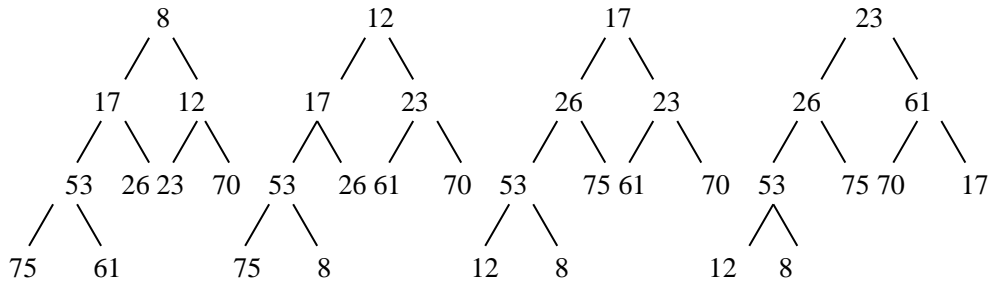
$H(12) = 12$ ,  $H(23) = 10$ ,  $H(45) = 6$ ,  $H(57) = 5$ ,  $H(20) = 7$ ,  $H(03) = 03$ ,  
 $H(78) = 0$ ,  $H(31) = 5$ ,  $H(15) = 2$ ,  $H(36) = 10$

0	1	2	3	4	5	6	7	8	9	10	11	12
78		15	03		57	45	20	31		23	36	12
1	1	1		1	1	1	4		1	2		1

$ASL_{succ} = 14/10 = 1.4$

得分

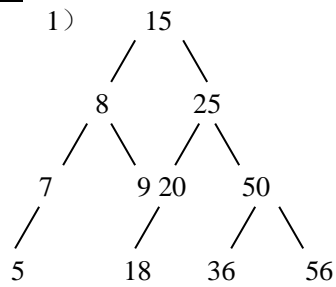
5. (10 分)



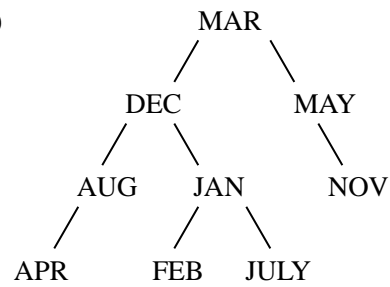
得分

6. (10 分)

1)



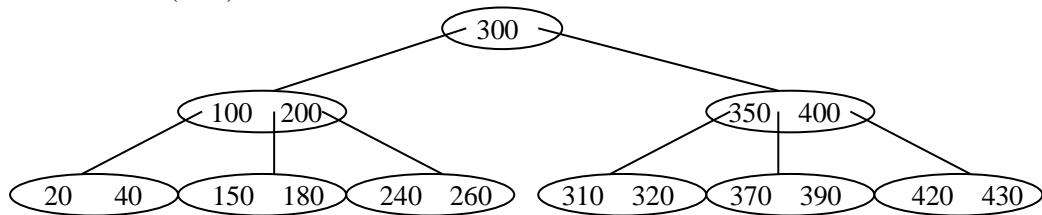
2)



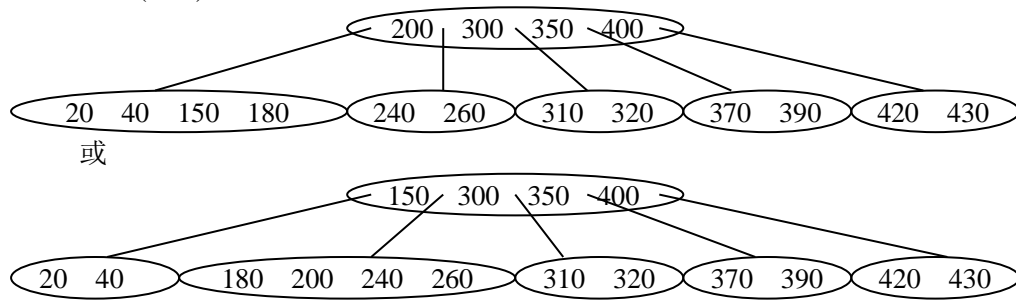
得分

7. (10 分)

插入 390 (5 分)



删除 100 (5 分)



得分

8. (10 分)

```
1 rear = rear . link
2 rear . link = p . link
3 p . link = rear ;
```

得分

9. (10 分)

```
class BinaryTree;
class BinTreeNode
{ friend class BinaryTree;
private:
    BinTreeNode * left, *right ;
    Int data;
}
class BinaryTree
{ public:
    int isBST(BinTreeNode *T);
private:
    BinTreeNode * root;
}
int BinaryTree :: isBST(BinTreeNode * T)
{ if ( (T == NULL) || (T->left == NULL && T->right == NULL)) return 1;
  else if ( T->left == NULL) return (isBST(T->right) && (T->data < min(T->right)));
  else if ( T->right == NULL) return (isBST(T->left) &&
                                     (T->data > max(T->left)));
  else return (isBST(T->left) && isBST(T->right) &&
               T->data >max( T->left)&& T->data < min(T->right));
}
```

Max(t) 函数的返回值为求 t 树的最大值;

Min(t) 函数的返回值为求 t 树的最小值