

1. 填空题 (36 分, 每空 3 分)

1) 直接插入排序:  $n-1$  。快速排序:  $n*(n-1)/2$  。  
折半插入排序, 直接选择排序

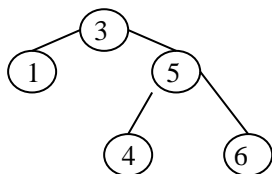
2) 6

3) 3

4) 0

5)  $(n*(n-1))/2$  无向完全图

6)

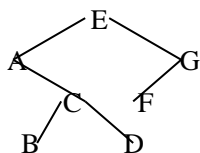


$k_1 = 1$

$k_2 = 5$

$k_3 = 7$

7) EACBDGF



8) 最大堆为 92, 86, 56, 70, 33, 33, 48, 65, 12, 24

2. 解答题 (40 分, 每题 10 分)

1)  $H(26) = 26 \% 17 = 9$        $H(8) = 8 \% 17 = 8$   
 $H(25) = 25 \% 17 = 8$        $H(18) = 18 \% 17 = 1$   
 $H(72) = 72 \% 17 = 4$        $H(59) = 59 \% 17 = 8$   
 $H(38) = 38 \% 17 = 4$

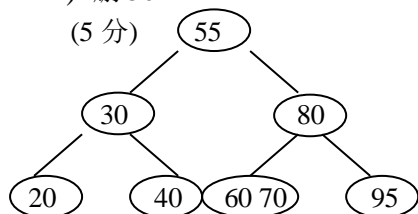
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
	18			72	38			25	26	8	59					
	1			1	2			1	1	3	4					

1) 59 的散列地址为 11

2) 搜索了 4 次

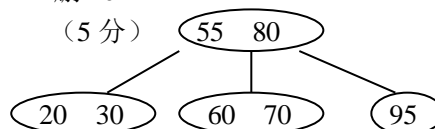
2) 删 50

(5 分)



删 40

(5 分)

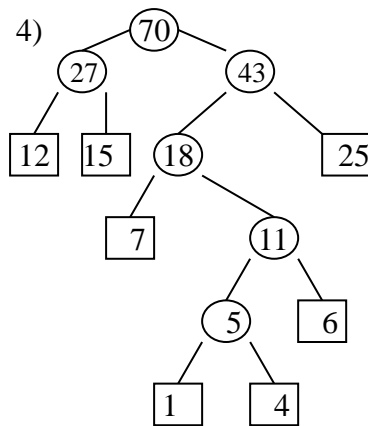


3) 1 2 25

1 2 4 41

1 2 4 3 48

1 2 4 3 5 53



$$(12+15+25)*2+7*3+6*4+(1+4)*5 = 174$$

### 3 算法题 (24 分, 第 1 题 10 分, 第 2 题 14 分)

```
1) class ListNode
{
    int data;
    ListNode link;
}
float Average( ListNode f, int n)           1 分
{
    if ( f . link  ==  NULL ) return f . data ;           3 分
    else return ( Average ( f . link , n-1 ) * ( n-1 ) + f . data ) / n ;           6 分
}
```

```
2) class BinaryNode
{
    ; //构造函数
    int data;
    BinaryNode left;
    BinaryNode right;
}                                           2 分

public class BinarySearchTree
{
    public BinarySearchTree( ) { root = null; }

    .

    public void BSTFind( BinarySearchTree t, int k );
    private BinaryNode root ;
}                                           2 分

public void BSTFind(BinarySearchTree t, int k)
{
    if (t)                                           3 分
    {
        BSTFind(t.right , k);                       2 分
        if (t.data>=k) System.out.println( t.data ); 3 分
        BSTFind(t.left , k);                         2 分
    }
}
```