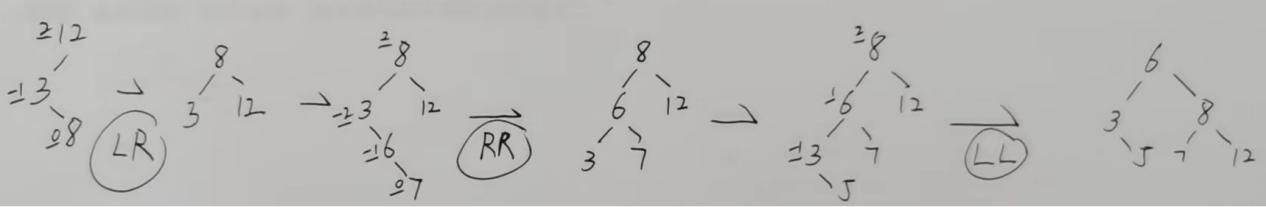
一、单项选择题（本大题共10小题，每小题2分，总计20分）

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| C | D | A | C | D | A | C | C | D | B |

二、简析题（本大题共2小题，总计10分）

1.（共4分）快速排序的原理是每次选择一个基准，将无序序列分为两段，左边的关键字小于基准，右边的大于基准，通过不断递归划分在理想情形下可形成一棵高度为log2n的二叉树，这样乘以划分次数n-1，总的时间复杂度即为O(nlog2n)；但如果每次找的基准都使得被划分的两部分中有一部分是空，即会形成一棵单支树，造成时间复杂度为最坏情况的O(n2)。（3分）；画图举例说明（1分）

2.（共6分）LR-RR-LL（4分）；画图说明（2分）



三、综合题（本大题共8小题，总计40分）

1.（共5分）

(1)后根序遍历序列:DHIJEBFKLGMNCA （2分）

(2)（2分）

先根序遍历序列:ABDEHIJCFGKLMN （1分）

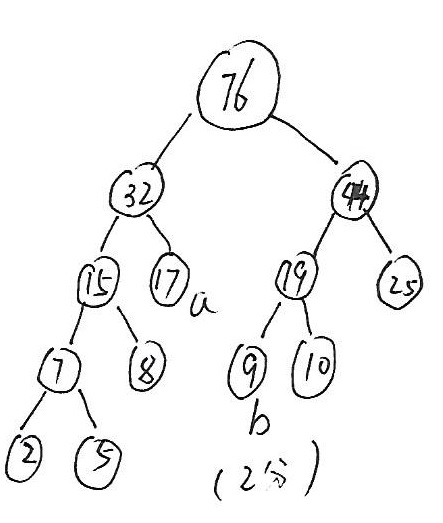
2.（共5分）

（1）（3分）

（2）ASL=2.5（2分）

3.（共5分）

（1）



（2）wpl=193 （1分）

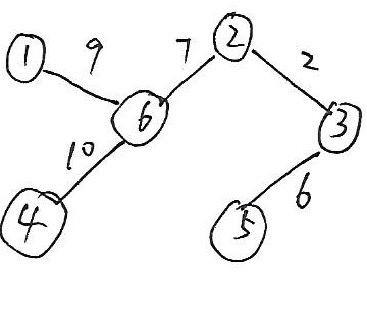
（3）a编码 01；b编码100 （2分）

4.（共5分）

（1）DFS 3,1,2,4,8,5,6,7 （3分）

（2）BFS 4,2,8,1,5,6,7,3 （2分）

5.（共5分）



要求画出步骤（5分）

6.（共5分）

（1）1,2,3,5,4,6,7,8,9 （2分）

（2）拓扑排序中找不到没有前驱的顶点，无法继续拓扑排序，称为死锁。（1分）

（3）拓扑排序算法中，发现所有顶点均有前驱，从而发现图中存在死锁。（2分）

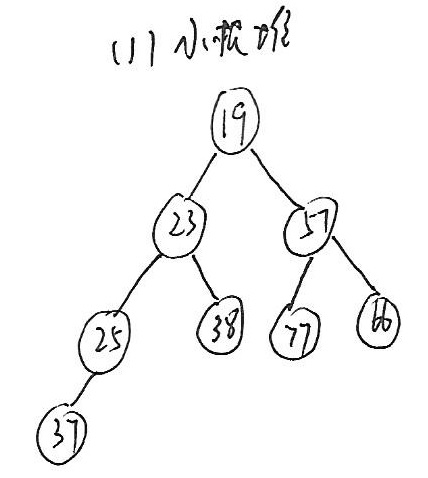
7.（共5分）

（1）26,37,15,22,67,56,32,79 （1分）

（2）15,26,67,37,22,79,56,32 （2分）

（3）32,26,22,15,37,79,56,67 （2分）

8.（共5分）

（2分）

（2）第一轮 23,25,57,19,38,77,66,37

第二轮 23,19,38,25,57,37,66,77 （3分）

**四、算法及开放性问答题（本大题共3小题，每题10分，总计30分）**

1. （10分）

参考答案：

void fun(NODE \*h, int x) //插入新结点并保持递增顺序

{

NODE \*p, \*q, \*s;

s = (NODE \*)malloc(sizeof(SLIST));

s->data = x ; /

q = h;

p = h->next;

while (p != NULL && x > p->data)

{

q = q->next;

p = p->next;

}

s->next = p;

q->next = s ; /

}

2.（10分）

（1）（5分）假设用线性探测法解决冲突，用除留余数法（除数p=9）作为哈希函数，请问上述哈希表进行查询的ASL是多少？

哈希函数：Hash(key)=key mod 9

线性探测法处理冲突：Hi=（Hash(key)+di）mod 9 di=1,2,3,...（1分）

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| 18 |  | 11 | 12 | 3 | 13 | 24 | 7 | 35 |

Hash(3)=3 冲突 （Hash(3)+1）mod 9=4

Hash(13)=4冲突 （Hash(13)+1）mod 9=5

（3分）

ASL=（6+2+2）/8=5/4（1分）

（2）（2分）请举例说明何为堆积现象？

如原本散列地址不是3关键字通过线性探测法获得了地址3，则后续实际散列地址为3的关键字将无法放入，只能继续进行线性探测，获取新的地址，因此散列地址不同的关键字，争夺同一个后继位置使得非同义词也加入了探测序列。

（3）（3分）你认为装填因子*a*对堆积现象有什么影响？

答案：装填因子过大，有可能会使堆积加剧.

3.（共10分）

（1）（6分）如主关键字的大小分布随机，且要求排序稳定，你认为归并排序、快速排序、冒泡排序，哪一种最适用于本问题？请完整描述你选用的算法，以及它的时间复杂度？

答案：

归并排序（2分）

归并排序的基本思想：将有序的子序列进行反复归并，得到有序序列。

对于二路归并排序：设初始序列含有n个记录，则可看成n个有序的子序列，每个子序列长度为1，然后两两合并，得到⎣n/2⎦个长度为2或1的有序子序列，再两两合并，重复此过程，直至得到一个长度为n的有序序列为止。（2分）

二路归并排序的时间复杂度是O(nlog2n) （2分）

（2）（4分）如果D中的数据元素含两个关键字，分别是k1，k2，要求先按k1排序，在保持k1有序的情况下，再按k2排序，请列举你认为最合适的两种对k2排序的算法，并简述原因？

归并排序，直接插入排序 （2分）

按K1排序后，K2可能按K1被分成关键字较少的不同分组，可选择直接插入排序或归并排序（2分）