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 | A | A | C | C | A | C | D | C | B |

二、简析题（本大题共2小题，总计10分）

1.（共4分）二叉树在单支树形态时用顺序存储性能最差，因为需要耗费更多存储空间（3分）；画图举例说明（1分）

2.（共6分）如果对元素个数已知，可采用多个顺序栈共享空间的方式，可充分利用顺序存储空间，但对每个栈顶指针的控制相对复杂（2分）；若对元素个数未知，可采用链栈的方式，可根据情况灵活分配存储空间，但由于指针变量也加大了空间开销（2分）；对于独立的顺序栈，栈操作操作相对简单，但同时由于元素个数未知，有可能会造成预先分配的空间过多或不足的问题（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:DHIJEBFGCA （2分）

(2)（2分）

先根序遍历序列:ABDEHIJCFG （1分）

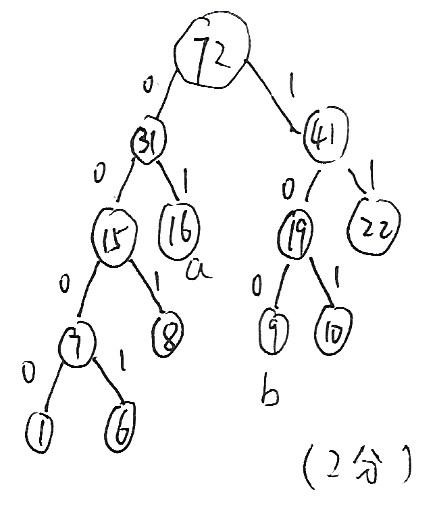
2.（共5分）

（1）（3分）

1. ASL=2.5（2分）

3.（共5分）

（1）



（2）wpl=185 （1分）

（3）a编码 01；b编码100 （2分）

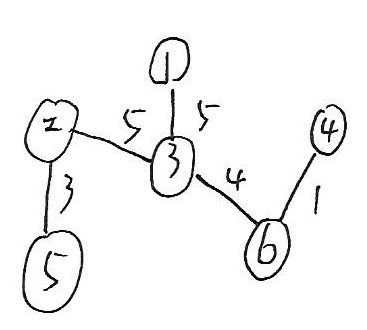
4.（共5分）

（1）DFS 5,7,6,3,4,2,1 （2分）

（2）BFS 2,1,3,5,7,4,6 （2分）

（3）DFS和BFS算法防止遍历时顶点“无重复，无遗漏”：通过数组标识已访问的顶点，已经访问过的不再访问，实现无重复；通过从每个顶点出发进行遍历实现无遗漏。（1分）

5.（共5分）



要求画出步骤（5分）

6.（共5分）

（1）1,2,3,6,5,4,7,8,9,10 （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）

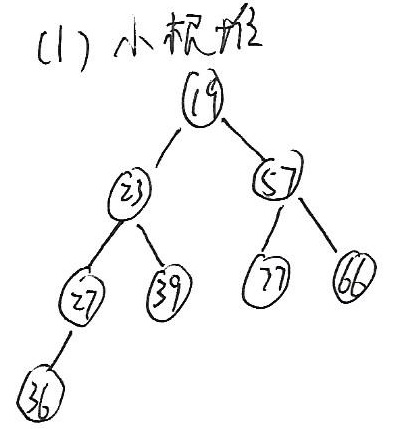
7.（共5分）

（1）25,38,18,21,65,59,33,78 （1分）

（2）18,25,65,38,21,78,59,33 （2分）

（3）33,25,21,18,38,78,59,65 （2分）

8.（共5分）

（2分）

（2）第一轮 23,27,57,19,39,77,66,36

第二轮 23,19,39,27,57,36,66,77 （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. （共10分）

void Del\_x(SList &L,){

int k = 0; //记录值不等于0的元素个数

for(int i = 0;i < L.len;i++){

if(L.data[i] != 0){

L.data[k] = L.data[i];

k++;

}

}

L.len = k;

}

2. （共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 |
| 9 | 27 |  | 12 | 13 | 23 | 15 | 25 | 68 |

Hash(27)=0 冲突 （Hash(27)+1）mod 9=1

Hash(68)=5 冲突 （Hash(68)+1）mod 9=6 冲突

（Hash(68)+2）mod 9=7冲突 （Hash(68)+3）mod 9=8 （3分）

ASL=（6+2+4）/8=1.5（1分）

（2）（3分）请举例说明什么是哈希地址冲突？

哈希冲突是指在使用哈希表存储数据时，两个或多个不同的键被哈希函数映射到同一个位置的情况,如本题中的关键字9和27，其哈希地址均为0。

（3）（2分）请列举至少两种解决冲突的技术。

答案：

线性探测法

链地址法

3.（共10分）

（1）（6分）如主关键字的大小分布随机，且不要求排序稳定，你认为归并排序、堆排序、快速排序， 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：

快速排序 （2分）

***算法描述***（2分）

***首先*：**对一组无序的数据集合，选择任意元素作为基准点，使得：

该基准点左边的所有记录都小于或等于它

使基准点右边的所有记录都大于或等于它

此时，该基准点将数据集合划分成两个半区

***然后：***分别对左,右半区进行快速排序

时间复杂度（2分）

最坏情况：当待排序序列有序时，每次划分的基准都是当前无序区中关键字最小(或最大)的记录，时间复杂度为O(*n*2)

平均情况为O(*nlog*2*n*)

（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？

答案：

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）