

中国大学 MOOC-陈越、何钦铭-数据结构-2018 春期末考试

一、判断题

1-1 若用平方探测法解决冲突，则插入新元素时，若散列表容量为质数，插入就一定可以成功。(2分)

F

1-2 对 N 个不同的数据采用冒泡排序进行从大到小的排序，当元素基本有序时交换元素次数肯定最多。(2分)

F

1-3 $n!$ 是 $O(n^n)$ 的。(2分)

T

1-4 对一棵平衡二叉树，所有非叶结点的平衡因子都是 0，当且仅当该树是完全二叉树。(2分)

F

1-5 无向连通图至少有一个顶点的度为 1。(2分)

F

二、选择题

2-1 对一组数据{2, 12, 16, 88, 5, 10}进行排序，若前三趟排序结果如下：第一趟排序结果：2, 12, 16, 5, 10, 88 第二趟排序结果：2, 12, 5, 10, 16, 88 第三趟排序结果：2, 5, 10, 12, 16, 88 则采用的排序方法可能是：(2分)

冒泡排序

2-2 给定有权无向图的邻接矩阵如下，其最小生成树的总权重是：(4分)

0	4	10	3	2
4	0	9	5	6
10	9	0	8	7
3	5	8	0	1
2	6	7	1	0

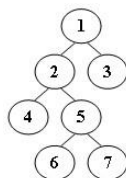
图 1

14

2-3 数据结构中 Dijkstra 算法用来解决什么问题？(2分)

最短路径

2-4 给定二叉树如下图所示。设 N 代表二叉树的根， L 代表根结点的左子树， R 代表根结点的右子树。若遍历后的结点序列为 3、1、7、5、6、2、4，则其遍历方式是：(2分)



批注 [张1]: 课本上说了需要满足某个 $4k+3$ 形式的素数。举个反例假设容量为 5，已经有三个元素，插入一个元素，就不行了。

批注 [张2]: 完全无序时交换次数最多。

批注 [张3]: 要实现所有非叶结点的平衡因子都是 0，所有的叶结点必须是填满的，因此是完美二叉树。

批注 [张4]: 只有一个顶点的。

批注 [张5]: 采用冒泡法：2 与 12, 12 与 16, 16 与 88 之间不需要互换，但是 5 和 88, 10 和 88 之间需要互换，所以变为了 2, 12, 16, 5, 10, 88；第二趟，2 与 12, 12 与 16 不需要换，16 与 5, 10 与 10 需要换，得到 2, 12, 5, 10, 16, 88。第三趟以此类推。

批注 [张6]: 关于本题，我做出来的权重是 15，走的路径是 3-5-4-1-2，即 $7+1+3+4=15$ 。待求证。

批注 [张7]: 典型的右中左访问方式，因此是 RNL。

图 2

RNL

2-5 设栈 S 和队列 Q 的初始状态均为空，元素{1,2,3,4,5,6,7}依次进入栈 S。若每个元素出栈后立即进入队列 Q，且 7 个元素出队的顺序是{2,5,6,4,7,3,1}，则栈 S 的容量至少是：(2 分)

4

解析：

S (FILO)	Q(FIFO)
1,2	
1,	2
1,3,4,5	2
1,3,4	2,5
1,3,4,6	2,5
1,3,4	2,5,6
1,3	2,5,6,4
1,3,7	2,5,6,4
1,3	2,5,6,4,7
1	2,5,6,4,7,3
	2,5,6,4,7,3,1

故容量至少为 4。

2-6 对于序列{49, 38, 65, 97, 76, 13, 27, 50}，按由小到大进行排序，下面哪一个是初始步长为 4 的希尔排序法第一趟的结果？(4 分)

49,13,27,50,76,38,65,97

批注 [张8]: 排序选中的第一个数字是 49，相比较的是 76，因为 49 小于 76，故位置不变；38 和 13 比较，13 与 38 交换位置……以此类推，得出答案。

2-7 在一个有权无向图中，如果顶点 b 到顶点 a 的最短路径长度是 10，顶点 c 与顶点 b 之间存在一条长度为 3 的边。那么下列说法中有几句是正确的？(4 分)

1. c 与 a 的最短路径长度就是 13
2. c 与 a 的最短路径长度就是 7
3. c 与 a 的最短路径长度不超过 13
4. c 与 a 的最短路径不小于 7

批注 [张9]: 画图可知，c 与 a 的最短路径可以是 13 也可以是 7，故第一句和第二句话不对。由于 b 到 a 的最短路径是 10，故第三句第四句话正确。所以 2 句正确。具体自己画图就明白了。

2 句

2-8 在拓扑排序算法中用堆栈和用队列产生的结果会不同吗？(2 分)

有可能会不同

批注 [张10]: 我记得视频有讲过，记住即可。

2-9 设最小堆（小根堆）的层序遍历结果为{5,18,15,28,22,42,40}。用线性时间复杂度的算法将该堆调整为最大堆（大根堆），则该树的中序遍历结果为：(4 分)

18,28,22,42,15,40,5

批注 [张11]: 因为层序遍历，所以其实画出最小堆的图来非常容易。然后由于线性时间复杂度的算法，所以就是思路最简单的那个，然后就可以调整出最大堆来了。然后根据左中右的顺序读出即可。

2-10 在图中自 a 点开始进行广度优先遍历算法可能得到的结果为：(2 分)

批注 [张12]: BFS 算法，对于 a 来说，路径长度为 1 的点是 b,c,e，路径长度为 2 的点是 d,f。根据这一点就可以得出排序结果来了。

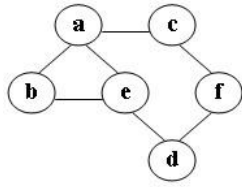


图 3

a,b,e,c,d,f

2-11 在一个链队列中，front 和 rear 分别为头指针和尾指针，则插入一个结点 s 的操作为 ()。(2 分)

rear->next=s;rear=s;

2-12 设散列表的地址区间为[0,16]，散列函数为 $H(Key)=Key\%17$ 。采用线性探测法处理冲突，并将关键字序列{26, 25, 72, 38, 8, 18, 59}依次存储到散列表中。元素 59 存放在散列表中的地址是：(4 分)

11

解析：

26/17—9 25/17—8 72/17—4 38/17—4 8/17—8 18/17—1 59/17—8

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

$(h(38)+1)\%17=5$ $(h(8)+1)\%17=9$ 由于 9 已经被填满，所以 $(h(8)+1)\%17=10$

由于 8,9,10 已经被填满，所以 $(h(59)+3)\%17=3$

2-13 令 P 代表入栈，O 代表出栈。则将一个字符串 $3*a+b/c$ 变为 $3a*bc/+$ 的堆栈操作序列是哪个？（例如将 ABC 变成 BCA 的操作序列是 PPOPOO。）(4 分)

POPPOOPPOPPPOO

2-14 要判断一个整数 N (>10) 是否素数，我们需要检查 3 到 \sqrt{N} 之间是否存在奇数可以整除 N 。则这个算法的时间复杂度是：(2 分)

$O(\sqrt{N})$

2-15 将 1~6 这 6 个键值插到一棵初始为空的二叉搜索树中。如果插入完成后，搜索树结构如图所示，问：可能的插入序列是什么？(2 分)

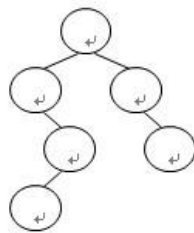


图 4

413256

批注 [张13]: 想想一下就可以理解。

批注 [张14]: 解析见答案下方。

批注 [张15]: 这个按照课本 P82 的原则推导一下即可。

批注 [张16]: 显然可知。

批注 [张17]: 所给 4 个选项的顺序画出二叉搜索树，然后发现只有 413256 的顺序符合要求。

2-16 将 9,8,7,2,3,5,6,4 顺序插入一棵初始为空的 AVL 树。下列句子中哪句是错的? (4 分)

5 是根结点

2-17 对给定序列{110, 119, 7, 911, 114, 120, 122}采用次位优先 (LSD) 的基数排序, 则两趟收集后的结果为: (2 分)

7,110,911,114,119,120,122

解析:

最次位排序:

序号	0	1	2	3	4	5	6	7	8	9
FIRST	110	911	122		114			007		119
	120									
SECOND	007	110	120							
		911								
		114	122							
		119								

故得出答案的排序。

2-18 给定输入序列{4371,1323,6173,4199,4344,9679,1989}以及散列函数 $h(X)=X\%10$ 。如果用大小为 10 的散列表, 并且用分离链接法解决冲突, 则输入各项经散列后在表中的下标为: (-1 表示相应的插入无法成功) (4 分)

1,3,3,9,4,9,9

2-19 在图中自 d 点开始进行深度优先遍历算法可能得到的结果为: (2 分)

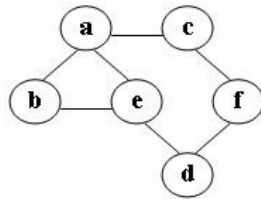


图 5

d,e,a,c,f,b

2-20 哈夫曼树是 n 个带权叶子结点构成的所有二叉树中 (带权路径长度) 最小的二叉树。 (2 分)

2-21 在并查集问题中, 已知集合元素 0~8 所以对应的父结点编号值分别是{1,-4,1,1,-3,4,4,8,-2} (注: -n 表示树根且对应集合大小为 n), 那么将元素 6 和 8 所在的集合合并 (要求必须将小集合并到大集合) 后, 该集合对应的树根和父结点编号值分别是多少? (4 分)

4 和-5

解析:

元素	0	1	2	3	4	5	6	7	8
信息	1	-4	1	1	-3	4	4	8	-2

结构如下图所示:

批注 [张18]: AVL 是平衡二叉树, 即要求平衡因子为 0、1 或 -1。当根结点为 5 时, 其平衡因子不符合上述条件。

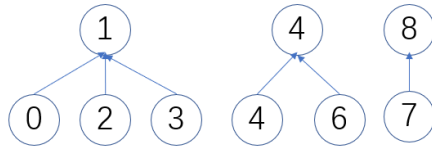
批注 [张19]: 见答案下面解析。

批注 [张20]: 散列查找 P183, 类似于 2-12, 只不过相同 h(key)值的数用链表连接在同一个下标下面。

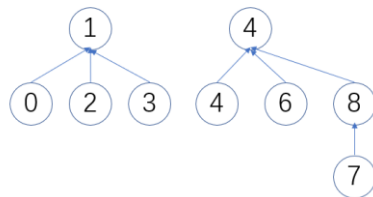
批注 [张21]: DFS 类似于 BFS。与 d 路径长度为 1 的点有 e 和 f, 按照字母顺序排序先访问 e。由于与 e 路径长度为 1 的点有 b 和 a, 按照字母顺序先访问 a, 然后与 a 路径长度为 1 的是 c, 与 c 路径长度为 1 的是 f, 访问完后访问 b, 得到 deacfb。

批注 [张22]: 书上定义。

批注 [张23]: 见答案下面的解析。



合并后的情况如下：



自然也就是这样的情况了：

元素	0	1	2	3	4	5	6	7	8
信息	1	-4	1	1	-5	4	4	8	4

因此答案为 4 和-5。

2-22 将 10,12,1,14,6,5,8,15,3,9,7 逐个按顺序插入到初始为空的最小堆中，然后连续执行两次删除最小元素操作（DeleteMin），再插入 4，16，此后堆顶的元素是什么？（4 分）

4

二、程序填空题

1. 下列代码的功能是将一列元素 `r[1] ... r[n]` 按非递减顺序排序。普通选择排序是每次仅将一个待排序列的最小元放到正确的位置上，而这个另类的选择排序是每次从待排序列中同时找到最小元和最大元，把它们放到最终的正确位置上。

```
void sort( list r[], int n )
{
    int i, j, mini, maxi;

    for (i=1; i<n-i+1; i++) {
        mini = maxi = i;
        for( j=i+1; j<=n-i+1; ++j ){
            if( r[j]->key<r[mini]->key (3分) ) mini = j;
            else if(r[j]->key > r[maxi]->key) maxi = j;
        }
        if( mini != i (3分) ) swap(&r[mini], &r[i]);
        if( maxi != n-i+1 ){
            if( maxi == i (3分) ) swap(&r[mini], &r[n-i+1]);
            else swap(&r[maxi], &r[n-i+1]);
        }
    }
}
```

2. 下列代码的功能是将小顶堆 H 中指定位置 P 上的元素的整数键值下调 D 个单位，然后继续将 H 调整为小顶堆。

```

void DecreaseKey( int P, int D, PriorityQueue H )
{
    int i, key;
    key = H->Elements[P] - D;
    for ( i =  (3分); H->Elements[i/2] > key; i/=2 )
         (3分);
    H->Elements[i] = key;
}

```

三、编程题

7-1 还原二叉树 (8 分)

给定一棵二叉树的先序遍历序列和中序遍历序列，要求计算该二叉树的高度。

输入格式：

输入首先给出正整数 N (≤ 50)，为树中结点总数。下面两行先后给出先序和中序遍历序列，均是长度为 N 的不包含重复英文字母（区别大小写）的字符串。

输出格式：

输出为一个整数，即该二叉树的高度。

输入样例：

```

9
ABDFGHIEC
FDHGIBEAC

```

输出样例：

```
5
```

代码：

暂时不公布代码