

开始时间: 22:47

北京邮电大学 2020—2021 学年学期

《 数据结构 》 期末考试试题

考试 注意 事项	一、学生参加考试须带学生证或学院证明，未带者不准进入考场。学生必须按照监考教师指定座位就坐。 二、书本、参考资料、书包等物品一律放到考场指定位置。 三、学生不得另行携带、使用稿纸，要遵守《北京邮电大学考场规则》，有考场违纪或作弊行为者，按相应规定严肃处理。 四、学生必须将答题内容做在试题答卷上，做在试题及草稿纸上一律无效										
考试 课程	数据结构				考试时间			2020 年 12 月 19 日			
题号	一	二	三	四	五	六	七	八	九		总分
满分	20	10	10	10	10	10	10	10	10		
得分											
阅卷 教师											

一. 选择填空题 (20 分, 每空 2 分)

1. 以下术语，与数据的存储结构无关的是 (D)。
- A. 二叉链表 ✓ B. 循环队列 C. 哈希表 D. 有序表

2. 下列程序段的时间复杂度是 (A)。

count = 0;

for (k=1; k<=n; k*=2) log n

for (j=1; j<=n; j++) n

count ++;

- A. $O(n \log_2 n)$ B. $O(n)$ C. $O(\log_2 n)$ D. $O(n^2)$

3. 单链表中，增加一个头结点的目的是为了 (C)。

- A. 使单链表至少有一个结点 B. 标识表结点中首结点的位置

C. 方便运算的实现



D. 说明单链表是线性表的链式存储

4. 若一个栈的输入序列为 P_1, P_2, \dots, P_n , 其输出序列是 $1, 2, 3, \dots, n$, 若 $P_3=1$, 则 P_1 的值是 (B)。

A. 可能是 2 B. 不可能是 2 ~~C. 一定是 2~~ D. 不可能是 3

5. 设森林 F 共有 15 个分支, 25 个结点, 则 F 中树的个数是 (C)。

A. 8 B. 9 C. 10 D. 11

6. 具有 6 个顶点的无向图, 至少有 (D) 条边时就能确保是一个连通图。

A. 8 B. 9 C. 10 D. 11

7. 以下排序算法中, (C) 在一趟结束后不能保证有一个元素放在其最终位置。

A. 快速排序 B. 简单选择排序 C. 直接插入排序 D. 起泡排序

8. 完成在双循环链表结点 p 之后插入 s 的操作是 (D)。

A. $p \rightarrow next = s$; $s \rightarrow prior = p$; $p \rightarrow next \rightarrow prior = s$; $s \rightarrow next = p \rightarrow next$;

B. $p \rightarrow next \rightarrow prior = s$; $p \rightarrow next = s$; $s \rightarrow prior = p$; $s \rightarrow next = p \rightarrow next$;

C. $s \rightarrow prior = p$; $s \rightarrow next = p \rightarrow next$; $p \rightarrow next = s$; $p \rightarrow next \rightarrow prior = s$;

D. $s \rightarrow prior = p$; $s \rightarrow next = p \rightarrow next$; $p \rightarrow next \rightarrow prior = s$; $p \rightarrow next = s$;

9. 在关键字随机分布的情况下, 用二叉排序树的方法进行查找, 其平均查找长度与 (B) 数量级相当。

A. 顺序查找 B. 折半查找 C. 分块查找 D. 均不正确

10. 在存储数据时, 不仅需要存储各数据元素的值, 还要存储 (C)。

A. 数据的操作方法

B. 数据元素的类型

C. 数据元素之间的关系

D. 数据的存取方法

二. 判断题 (10 分, 每空 1 分)

1. (X) 数据的逻辑结构说明数据元素之间的顺序关系, 它依赖于计算机的存储结构。

2. (X) 在线性表中, 每个数据元素都有唯一的后继和唯一的前驱。

3. (X) 循环队列把存储空间想象成一个环, 因此它适合采用循环链表来存储。

4. (X) 哈夫曼树中可能有度为 1 的结点，也可能没有。
5. (✓) 折半查找的性能和二叉排序树的查找性能相同。
6. (✓) 排序过程中，简单选择排序的关键字比较次数与待排序列的初态无关。
7. (✓) (X) 若将存储结构从顺序存储改为链式存储，起泡排序的时间效率会降低。
8. (✓) 若完全二叉树的某个节点没有左子，也一定没有右子。
9. (✓) (X) 连通图上各边权值均不相同，则该图的最小生成树是唯一的。
10. () 快速排序算法不是稳定排序算法，其空间复杂度也不是 $O(1)$ 。

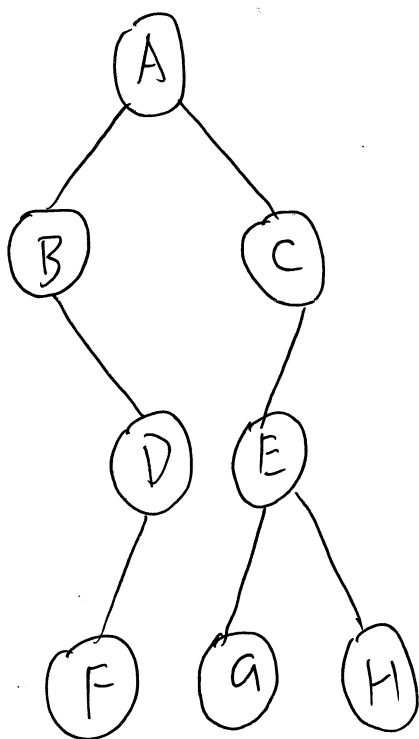
并非所有都有，无向图

快速排序算法

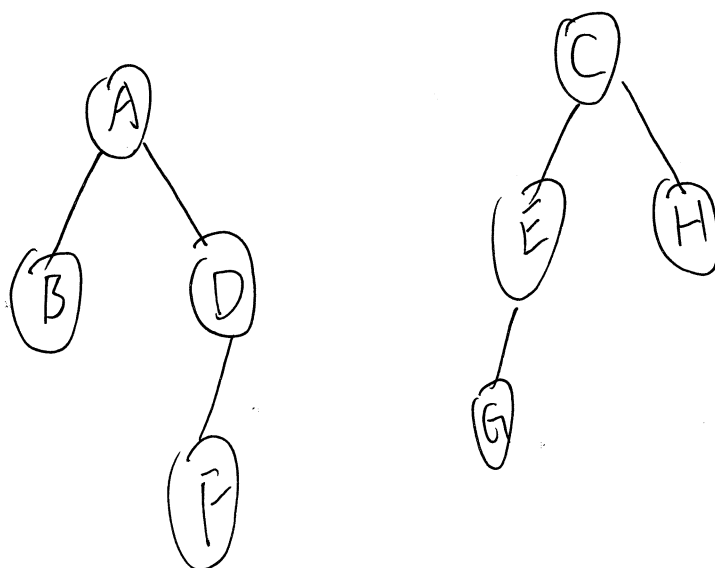
三. 已知一颗二叉树的先序序列和中序序列如下所示，画出这颗二叉树，并将其转换成对应的树或森林。(10 分)

先序序列: **ABDFCEGH**

中序序列: **BFDAGEHC**



~~树~~ 森林:



哈希复习 [第八章有题31]

284-15

269-11=24

38 264

23

四. 已知散列表的地址空间为 $A[0..10]$, 散列函数 $H(k) = (2k-15) \bmod 11$,

采用线性探测再散列法处理冲突。(10分)

68

57 110-91

72 182-15

(1) 请将关键字集合 {34, 19, 55, 91, 36, 82, 51, 77, 142, 10} 依次插入到下面的散列表中; 167

关键字	34	19	55	91	36	82	51	77	142	10
H(k)	9	1	7	2	2	6	10	7	5	5

164-15

149

162

87

154

139

散列地址	0	1	2	3	4	5	6	7	8	9	10
关键字	10	19	91	36		142	82	55	77	34	51
比较次数	7	1	1	2	1	1	1	1	2	1	1
查找失败	11	11	11	11	11	11	11	11	11	11	11

(2) 完成上表, 并计算出在等概率情况下查找成功和不成功时的平均查找长度。

ASL_{成功} = 1.7

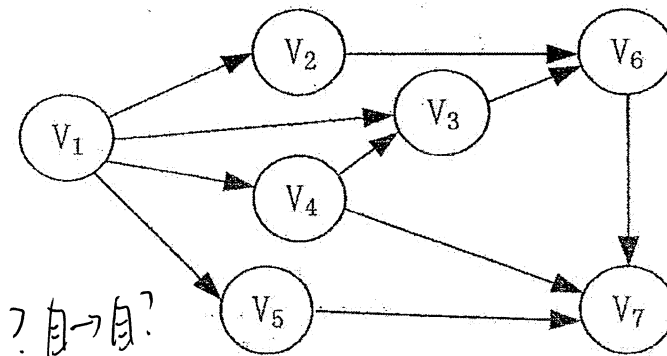
ASL_{不成功} = 1.4

五. 看下边的有向无环图，回答以下问题：（10 分）

(1) 画出它的邻接矩阵；（4 分）

(2) 从 V_1 出发按照上述邻接矩阵的存储结构，写出深度优先遍历的次序；（3 分）

(3) 写出可能的 3 个拓扑排序序列；（3 分）



(1)

	V_1	V_2	V_3	V_4	V_5	V_6	V_7
V_1	1	1	1	1	1	0	0
V_2	0	1	0	0	0	1	0
V_3	0	0	1	0	0	1	0
V_4	0	0	1	1	0	0	1
V_5	0	0	0	0	1	0	1
V_6	0	0	0	0	0	1	1
V_7	0	0	0	0	0	0	1

(2) $V_1, V_2, V_6, V_7, V_3, V_4, V_5$

(3) $V_1, V_2, V_4, V_3, V_6, V_7, V_5$
 $V_1, V_4, V_3, V_5, V_2, V_6, V_7$
 $V_1, V_5, V_2, V_4, V_3, V_6, V_7$

B.G.R.W.Y

六. 已知待排关键字序列: {Green, Blue, White, Red, Black, Yellow},
利用某种排序算法得到其第一趟排序结果如下所示, 请回答以下问题: (1)
请选择该排序算法是哪种排序方法, 并填入空格中; (2) 依照该排序算法原

理写出后续每趟排序的结果填入下表 (注意: 有几趟就写几趟)。(10 分)

第 1 趟排序: Black, Blue, Green, Red, White, Yellow

第 2 趟排序: _____

第 3 趟排序: _____

第 4 趟排序: _____

第 5 趟排序: _____

第 6 趟排序: _____

该排序算法为 快速 排序。

插入 X

选择 X

冒泡 X

希尔 ? X

堆 X

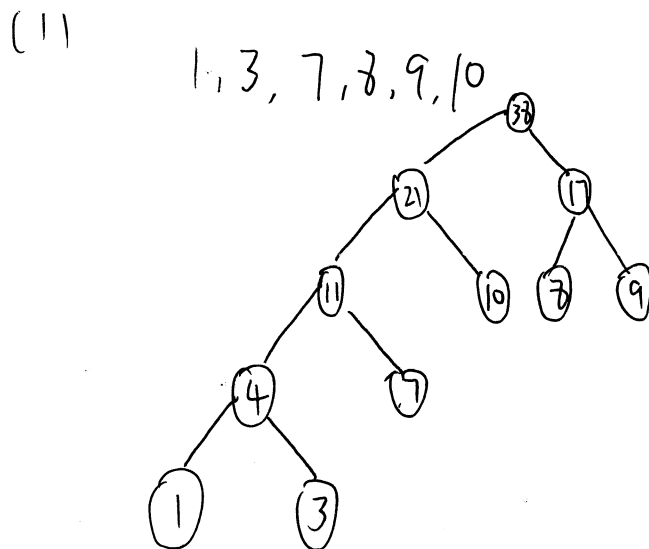
快排 ✓

归并排

七. 设 A、B、C、D、E、F 六个字母出现的频度分别为{8, 3, 9, 7, 10, 1};
回答以下问题:

(1) 构造相应的哈夫曼树, 请画出构造步骤;

(2) 求出此哈夫曼树的带权路径长度。(10 分)



(2) $WPL = 91$

- 八. 已知某门课考试成绩单包含姓名, 学号, 成绩三项, 已按照学号有序存储在带头结点的单链表中, 在此链表中每个结点增加一个空的指针域 (用 sort 表示), 用以指向排序后该结点的后继结点; 试给出算法, 在原单链表 next 指针不动的情况下, 按成绩递增的顺序建立该 sort 链。要求: 不得使用除该链表以外的任何链结点空间。(10 分)

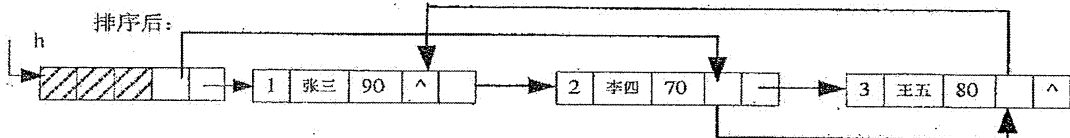
结点结构:

no	name	score	sort	next
----	------	-------	------	------

排序前:



排序后:



```
typedef struct node {
    int    no;
    char   name[10];
    int    score;
    struct node *sort, *next;
} node, *list;
```

```
void Sort(list &h) //h 为头指针
```

```
{
    h->score = 0;
    last score = 0;
    node* prev = h;
    node* min = NULL;
    Bool flag = 1;
    while(flag)
    {
        for(node* p = h->next, p, p = p->next)
        {
            if((min == NULL && p->score >= prev->score) || (min != NULL && p->score < min->score))
            if(p->prev && (min == NULL && p->score >= prev->score) || (min != NULL && p->score < min->score))
                min = p;
        }
        if(min != NULL)
        {
            min = min->next;
            prev->next = min;
            prev->next = min;
            prev = min;
            min = NULL;
        }
        else
        {
            prev->next = NULL;
            flag = 0;
        }
    }
}
```

分)

```
if (t → data == x)
```

第 10 页, 共 10 页