

东南大学考试卷 (A 卷)

课程名称 数据结构基础 考试学期 10-11-2 得分
 适用专业 计算机 考试形式 半开卷 考试时间长度 120 分钟

一、选择题 (在四个备选答案中, 选出一个最合适的答案, 每空 2 分, 共 30 分)

1、若无向图 G 的邻接矩阵如下:

$$\begin{matrix} & \textcircled{0} & \textcircled{1} & \textcircled{2} & \textcircled{3} \\ \begin{pmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{pmatrix} & \textcircled{0} \\ & \textcircled{1} \\ & \textcircled{2} \\ & \textcircled{3} \end{matrix}$$

则图 G 中包含的边数

(A) 3

(B) 4

(C) 7

(D) 8

2、当一个有 n 个顶点的有向图用邻接矩阵 A 表示时, 顶点 v_i 的度是 (D)。

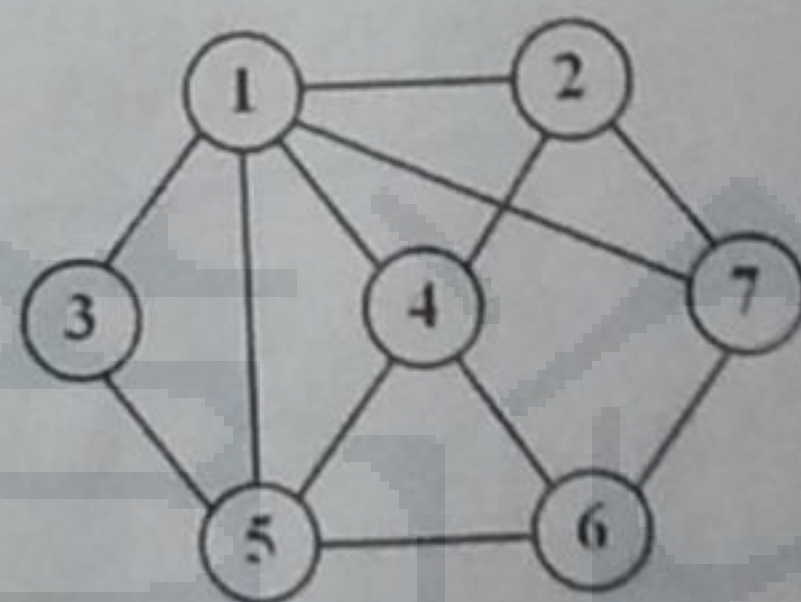
(A) $\sum_{i=1}^n A[i, j]$

(B) $\sum_{j=1}^n A[i, j]$

(C) $\sum_{i=1}^n A[j, i]$

(D) $\sum_{j=1}^n A[j, i] + \sum_{j=1}^n A[i, j]$

3、下图给出了 7 个顶点组成的无向图, 从顶点 1 出发对它进行广度优先遍历, 得到的顶点序列是 (C)



(A) 1534267

(B) 1726453

(C) 1354276

(D) 1247653

4、在有向图 G 的拓扑序列中, 若顶点 v_i 在顶 v_j 之前, 则下列情形不可能出现的是 (D)。

(A) G 中有弧 $\langle v_i, v_j \rangle$

(B) G 中有一条从 v_i 到 v_j 的路径

(C) G 中没有弧 $\langle v_i, v_j \rangle$

(D) G 中有一条从 v_j 到 v_i 的路径

5、任何一个带权无向连通图的最小生成树是 (C)

(A) 一定唯一的

- (B) 一定不唯一的
- (C) 有可能不唯一的
- (D) 有可能不存在的

6. 下列因素中, 影响排序算法稳定性关键因素是 (B)

- I. 待排序元素个数的多少
- II. 排序过程中是否发生了不相邻元素的交换
- III. 是否有关键码相同的元素
- IV. 排序算法是否采用递归方式实现

- (A) 仅 I
- (B) 仅 II
- (C) I 和 III
- (D) I 和 IV

7. 若数据元素序列: 11, 12, 13, 7, 8, 9, 23, 4, 5 是采用下述排序方法之一得到的第二趟排序后的结果, 则该排序算法只能是 (B)

- (A) 起泡排序
- (B) 直接插入排序
- (C) 选择排序
- (D) 二路归并排序

8. 对序列 15, 9, 7, 8, 20, 1, 4 用 Shell 排序方法进行排序, 若经一趟处理后的结果为 15, 1, 4, 8, 20, 9, 7, 则该趟采用的增量是 (D)

- (A) 1
- (B) 2
- (C) 3
- (D) 4

9. 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: 2, 12, 16, 5, 10, 88

第二趟: 2, 12, 5, 10, 16, 88

第三趟: 2, 5, 10, 12, 16, 88

则采用的排序方法可能是 (A)

- (A) 起泡排序
- (B) 希尔排序
- (C) 归并排序
- (D) 基数排序

10. 采用递归方式对顺序表进行快速排序, 下列关于递归次数的叙述中, 正确的是 (D)

- (A) 递归次数与初始数据的排列次序无关
- (B) 每次划分后, 先处理较长的分区可以减少递归次数
- (C) 每次划分后, 先处理较短的分区可以减少递归次数
- (D) 递归次数与每次划分后得到的分区处理顺序无关

11. 若对长度分别为 n 和 m 的两个有序序列进行二路归并, 则在归并过程中需要进行的最少和最多比较次数分别是 (C)

- (A) $\min(n, m)$, $n+m$
- (B) $\max(n, m)$, $n+m$
- (C) $\min(n, m)$, $n+m-1$
- (D) $\max(n, m)$, $n+m-1$

12. 为提高散列表的查找效率, 可以采取的正确措施是 (D)

- I. 增大装填 (载) 因子
- II. 设计冲突 (碰撞) 少的散列函数
- III. 处理冲突 (碰撞) 时避免产生聚集 (堆积) 现象

- (A) 仅 I
- (B) 仅 II
- (C) 仅 I、II
- (D) 仅 II、III

13. 设哈希表长为 14, 哈希函数是 $H(\text{key}) = \text{key} \% 11$, 表中已有数据的关键码为 15, 38, 61, 84 共四个, 现要将关键码为 49 的元素加到表中, 用二次探测法解决冲突, 则放入的位置是 (D)

(A) 3

(B) 5

(C) 8

(D) 9

14. 下面关于 m 阶 B 树的说法中, 正确的是(B)

- ① 每个结点至少有两棵非空子树
- ② 树中每个结点至多有 $m-1$ 个关键码
- ③ 所有叶子在同一层上
- ④ 当插入一个数据项引起 B 树结点分裂后, 树长高一层

(A) ①、②和③

(B) ②和③

(C) ②、③和④

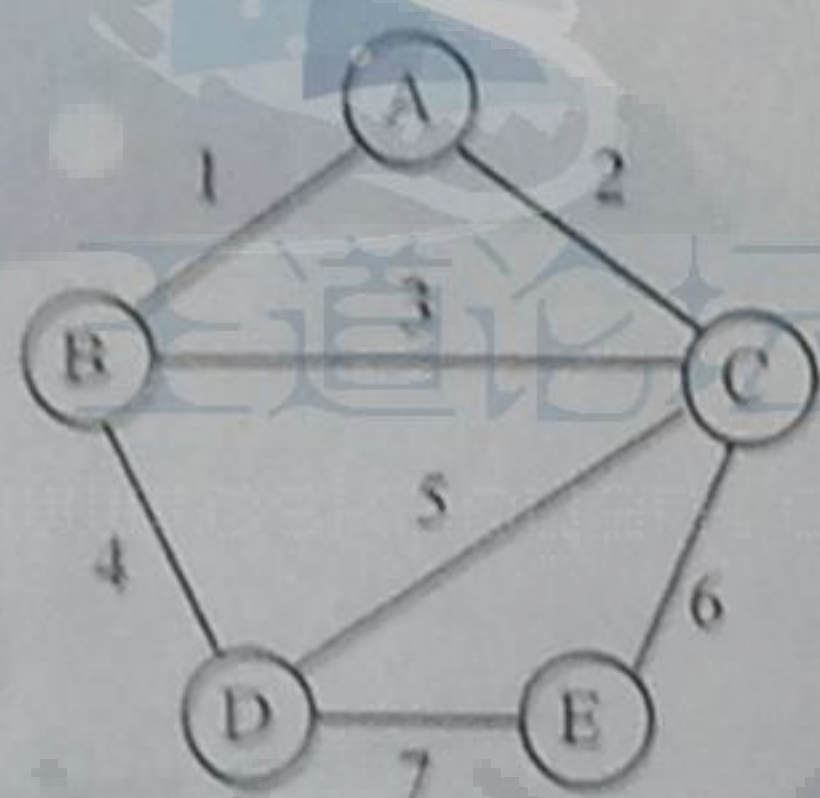
(D) ③

15. 下面关于 B 和 B⁺树的叙述中, 不正确的是(C)

- (A) B 树和 B⁺树都是平衡的多叉树
- (B) B 树和 B⁺树都可用于文件的索引结构
- (C) B 树和 B⁺树都能有效地支持顺序检索
- (D) B 树和 B⁺树都能有效地支持随机检索。

二、问答题 (供 10 题, 可选作其中的 8 题, 每题 7 分, 共 56 分)

1. 找图的最小生成树的过程, 就是在图中依次选择合适的边的过程。试分别用普里姆(Prim)算法 (从顶点 A 开始) 和克鲁斯卡尔 (Kruskal) 算法找下图的最小生成树, 按选择边的次序分别给出两个算法各自找出的最小生成树的边集合。



[参考解答]

选边的次序:

Prim: AB, AC, BD,

CE (4 分)

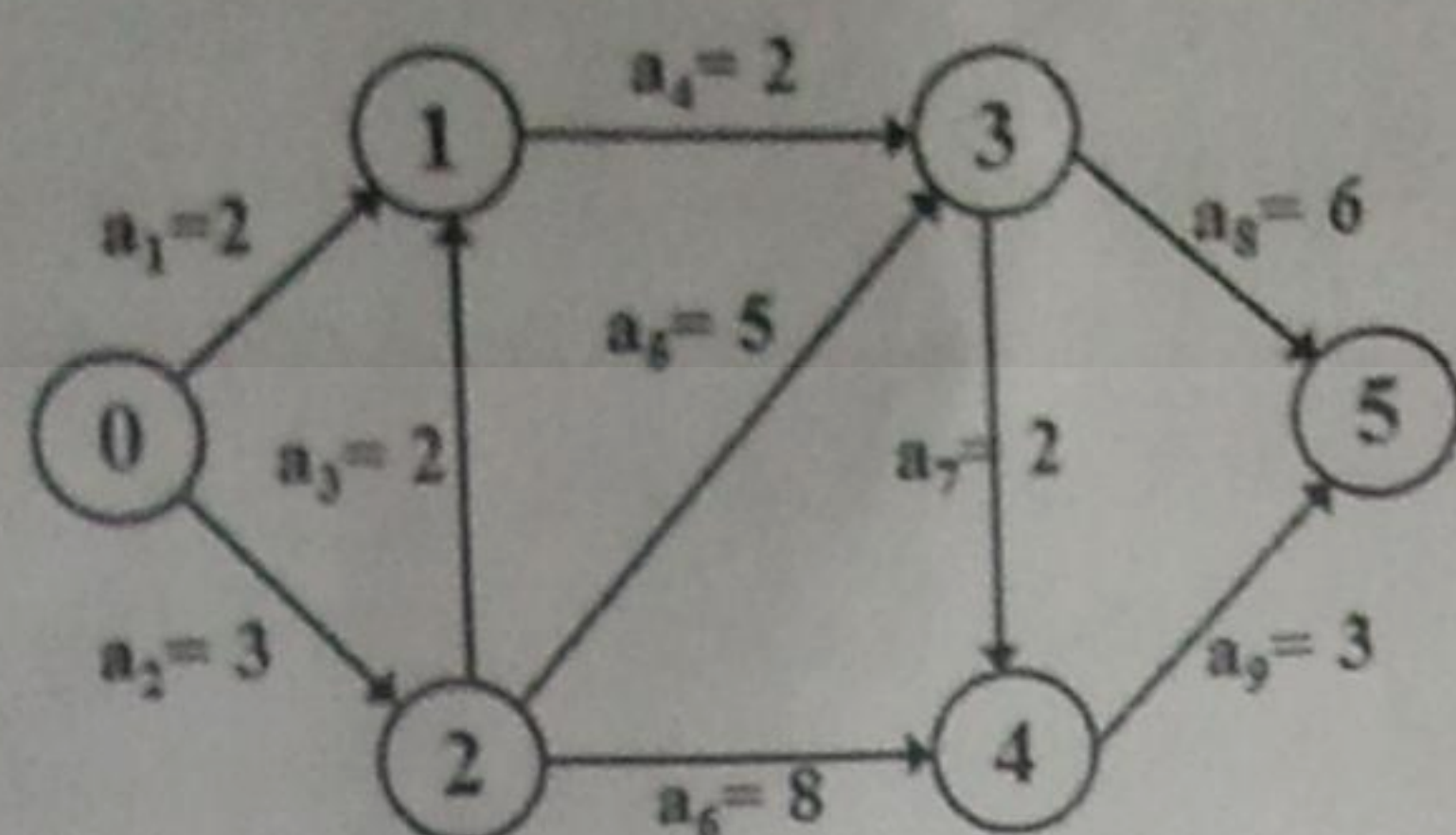
Kruskal: AB, AC, BD, CE (3 分)

2. 针对带权的 DAG 图 (有向无环图), 试提出一个求单源最短路径的 (不同于 Dijkstra 算法的) 方法 (只需简要说明该方法的基本思路, 不必程序实现), 并分析其时间复杂度。

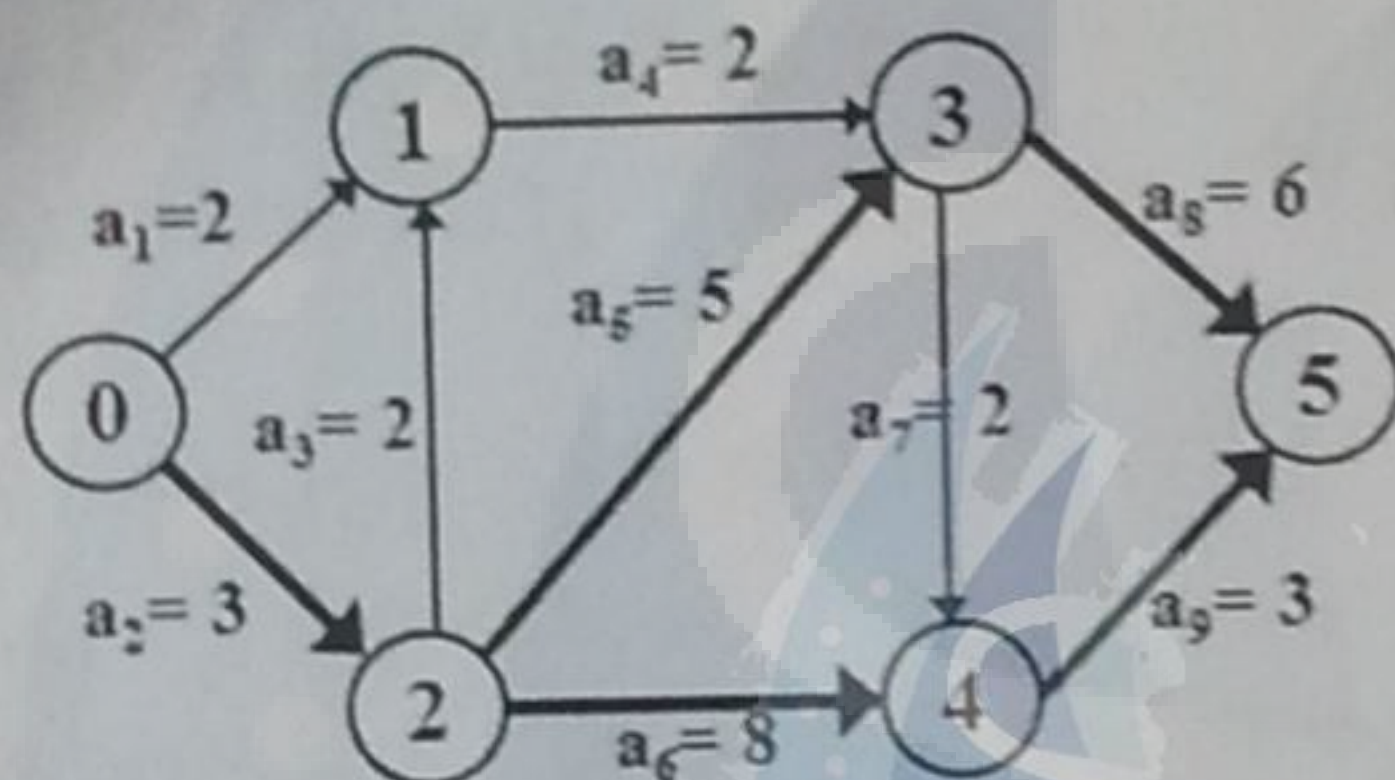
[参考解答]

利用拓扑排序的思想进行求解。按照拓扑排序的次序, 计算从源点到任一顶点 v 的最短路径时, v 的所有前驱顶点的最短路径都已经求出。该方法的时间复杂度为 $O(|N|+|V|)$ 。

3. 假设某工程用如下所示的 AOE 图表示。为了缩短整个工程的工期, 需要选择图中一条边, 缩短其持续时间。试问: 选择那条边最合适?



[参考解答]



图中的关键活动有： a_2 , a_5 , a_6 , a_8 , a_9 ，其中，缩短 a_2 表示的活动，可以缩短整个工程的工期。

4、外部排序时采用 k ($k > 2$) 路归并的方法，以此来减少归并的趟数从而提高效率，这种方法能否应用在内部排序（归并排序）中，从而也提高排序的效率？为什么？

[参考解答]

不能提高排序效率，最多达到 $O(n \log_2 n)$ 的时间复杂度（4分）（可以通过对时间复杂度的推导得出）（3分）。

5、对于下列内部排序算法：

- (A) 快速排序 (B) 直接插入排序 (C) 起泡排序
(D) 简单选择排序 (E) shell 排序

试回答以下问题：

- ① 其关键码的比较次数与待排序序列初始状态无关的算法有哪些？
- ② 不能保证每趟排序至少能将一个元素放到其最终位置上的排序算法有哪些？
- ③ 若数据表中有 N 个元素，若仅要求找出其中最小的 K 个元素 ($N \gg K$)，则最适合采用的排序算法有哪些？
- ④ 若待排序序列初始时已基本有序，则采用哪些排序算法效率最高？

[参考解答]

- ① 其元素比较次数与序列初始状态无关的算法有 (D) 简单选择排序。（1

分)

- ② 不能保证每趟排序至少能将一个元素放到其最终位置上的排序算法有 (B) 直接插入排序, (E) shell 排序。(2 分)
- ③ 若数据表中有 N 个元素, 现仅要求找出其中最小的 K 个元素 ($N \gg K$), 则最适合采用的排序方法是 (D) 简单选择排序, (C) 起泡排序。(2 分)
- ④ 若待排序序列初始时已基本有序, 则效率最高的排序算法是 (B) 直接插入排序, (C) 起泡排序。(2 分)

6. 基数排序的基本思想是: 将数据元素的关键码拆分成若干个子关键码, 然后按各个子关键码的取值范围, 分别进行“分配”和“收集”。现基于基数排序的思想, 采用某一基于比较的排序算法完成“分配”和“收集”过程, 即分别按子关键码进行排序, 最终完成基数排序。假设待排序元素关键码的取值是小于 1000 的正整数, 将关键码拆分为三个子关键码: “个位”、“十位”和“百位”。试问:

- ① 可采用哪个基于比较的排序算法对各个子关键码进行排序?
- ② 按什么次序先后对“个位”、“十位”和“百位”子关键码排序?

[参考解答]

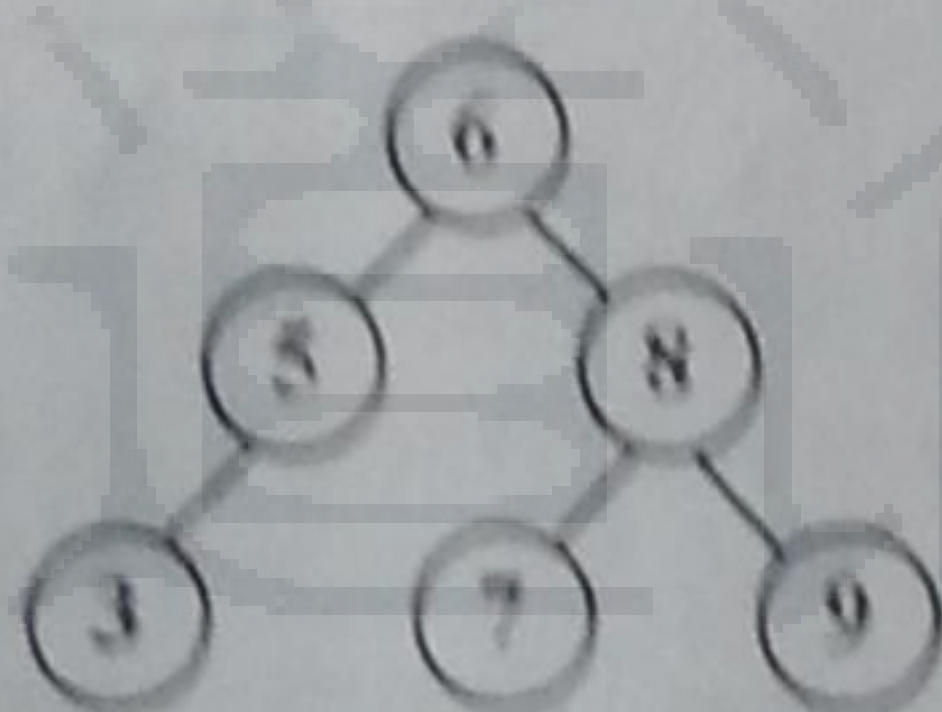
- ① 排序算法: 任何一个稳定的排序算法。(4 分)
- ② 排序次序: “个位”、“十位”和“百位”。(4 分)

7. 对一棵初始为空的 AVL 树, 若依次插入关键码为 9, 3, 5, 8, 6, 7 的结点, 对最后得到的 AVL 树, 试问:

- ① 根结点的关键码是什么?
- ② 树中有几个叶子结点? 其关键码各是什么?

[参考解答]

最后得到的 AVL 树如下所示:



- ① 根结点的关键码是 6。(3 分)
- ② 树中有 3 个叶子结点? 其关键码各是 3, 7, 9。(4 分)

8. 结点的插入和删除都是 AVL 树的基本操作。若在结点的插入或删除后发生了不平衡现象, 则需要进行调整。调整可通过“单旋转”或“双旋转”来实现。设 AVL 树的高度为 H , 试问: 最坏情况下, 一个结点的插入和删除各需进行多少次调整操作? 请说明原因。

[参考解答]

结点插入: 最多进行一次调整操作 (2 分)。因为结点插入时 AVL 树中的某个子树的调整, 不会影响其父结点的平衡状况; (1 分)

结点删除: 最多进行 $H-2$ 次调整操作 (2 分)。因为结点删除时 AVL 树

中的某个子树的调整,可能减小该子树的高度,因而影响其父结点的平衡状况。(2分)

9、设输入文件数据对象包含的关键码序列为: 14, 22, 7, 16, 11, 10, 12, 90, 26, 30, 28, 110。现采用置换-选择排序方法生成初始归并段,并假设内存工作区数组最多可容纳5个数据对象,请给出生成的初始归并段。

[参考解答]

第一段: 7, 10, 11, 12, 14, 16, 22, 26, 28, 30, 90, 110。

10、已知一组关键码为 (23, 15, 30, 22, 16), 利用除余法构造散列函数,散列函数为: $H(\text{key}) = \text{key} \% 7$ 。将这组关键码顺序插入散列表并用线性探查法解决冲突,散列表为 HT[10]。试画出构造出的散列表,并说明查找成功和查找不成功的平均查找长度分别是多少?

[参考解答]

key	23	15	30	22	16									
add	2	1	2	1	2									
	15	23	30	22	16									

$$ASL_{\text{succ}} = (1+1+2+4+4)/5 = 12/5$$

$$ASL_{\text{unsucc}} = (1+6+5+4+3+2+1)/7 = 22/7$$

① 元素放置位置正确 (4分)

② ASL_{succ} 正确 (2分), ASL_{unsucc} 正确 (1分)

三、算法设计题 (共2题, 可选做其中1题, 14分。请说明算法的设计思想)

1、几个驴友决定一起周游 n 个城市 ($0, 1, \dots, n-1$), 他们计划从城市 0 出发, 最后回到出发地, 为省钱他们不能重复到达任一城市 2 次, 即达到中间每个城市有且仅有 1 次。假设任意两个城市 i, j 之间交通费用 C_{ij} 都不相同, 他们计算如何使得总交通费最小是一件很困难的事, 刚给出一个方案很快又被另一个更好的方案推翻, 他们为采用什么方法计算交通费绞尽了脑汁, 最后决定干脆采用简单的贪婪法: 从 0 出发, 去最便宜的城市 A_1 ; 然后从 A_1 出发, 去最便宜的城市 A_2 ;; 达到 A_{n-1} 后回到 0。中间过程是除 0 外的 $n-1$ 个城市的一个全排列。请回答以下问题。

① 判断是否可以采用 Dijkstra 算法求最小交通费;

② 设计周游的城市序列的算法, 给出算法思想;

③ 分析出算法复杂度。

[参考解答]

① 不能采用 Dijkstra 算法。(2分)

② 定义如下矩阵描述城市之间的费用: (8分)

$$\begin{bmatrix} \infty & C_{0,1} & C_{0,2} & \cdots & C_{0,n-1} \\ C_{1,0} & \infty & C_{1,2} & \cdots & C_{1,n-1} \\ C_{2,0} & C_{2,1} & \infty & \cdots & C_{2,n-1} \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ C_{n-1,0} & C_{n-1,1} & C_{n-1,2} & \cdots & \infty \end{bmatrix}$$

从第 0 行中找到最小的元素 $C_{0,[1]}$ ，将第 0 行全置为 ∞ ；从第 [1] 行中找到最小的元素 $C_{[1],[2]}$ ，将第 [1] 行全置为 ∞ ；从第 [2] 行中找到最小的元素 $C_{[2],[3]}$ ，将第 [2] 行全置为 ∞ ；……；第 [n-2] 行中只剩下一个最小的元素 $C_{[n-2],[n-1]}$ 。(0,[1],[2],..., [n-1]) 就是所求的城市序列。

③ 算法复杂度为 $O(n^2)$ 。(4 分)

[评分标准]

- ① 若所设计的算法能够满足功能要求，给 8 分；
- ② 若所设计算法的思路正确，实现不完全正确，酌情给分；
- ③ 说明的时间复杂度若与所设计算法一致，给 4 分。

2. 在对 AVL 树进行查找的过程中，沿查找路径可产生一个关键码比较序列。如被查找元素在 AVL 树中，则此序列为从根结点到该元素路径上的关键码序列，否则为从根结点到某一叶结点路径上的关键码序列。试设计一个算法，判定任意一个给定的序列是否可能是对一棵 AVL 树进行查找的过程中产生的关键码比较序列，并请说明所设计算法的时间和空间复杂度。

[设计思想]

- ① 按序列的次序构造一棵二叉搜索树 BT，然后通过检查 BT 是否是单枝树来进行判定。算法的时间和空间复杂度分别为 $O(n)$ 和 $O(n^2)$ 。
- ② 对序列进行检查，任一元素的关键码是否都大于（或小于）右边的所有元素的关键码。算法的时间和空间复杂度分别为 $O(1)$ 和 $O(n^2)$ 。
- ③ 采用设上下限的方法从右向左扫描整个序列。算法的时间和空间复杂度分别为 $O(1)$ 和 $O(n)$ 。

[评分标准]

- ① 所设计的算法为①给 6 分；②给 8 分；③给 10 分；
- ② 若有其他算法，按①的时间和空间复杂度的同样标准给分；
- ③ 若所设计算法的思路正确，实现不完全正确，按①的标准酌情给分；
- ④ 说明的时间和空间复杂度若与所设计算法一致，各给 2 分。