

北京大学信息科学技术学院

考试专用纸

姓名：_____ 学号：_____ 授课教师：_____

考试科目：数据结构与算法 A 考试时间：2022 年 12 月 21 日 上午

题号	一	二	三	四	五	六	总分
分数							

北京大学考场纪律

1、考生要按规定的考试时间提前 5 分钟进入考场，隔位就座或按照监考人员的安排就座，将学生证放在桌面。无学生证者不能参加考试；迟到超过 15 分钟不得入场；与考试无关人员不得进入考场。如考试允许提前交卷，考生在考试开始 30 分钟后可交卷离场；未交卷擅自离开考场，不得重新进入考场继续答卷；交卷后应离开考场，不得在考场内逗留或在考场附近高声喧哗。

2、除非主考教师另有规定，学生只能携带必要的文具参加考试，其它所有物品（包括空白纸张、手机和智能手表等电子设备）不得带入座位；已经带入考场的手机和智能手表等电子设备必须关机，并与其他物品一起集中放在监考人员指定位置，不得随身携带或带入座位及旁边。

3、考试使用的试题、答卷、草稿纸由监考人员统一发放和收回，考生不得带出考场。考生在规定时间内答完试卷，应举手示意请监考人员收卷后方可离开；答题时间结束监考人员宣布收卷时，考生应立即停止答卷，在座位上等待监考人员收卷清点无误后，方可离场。

4、考生要严格遵守考场规则，在规定时间内独立完成答卷。不准旁窥、交头接耳、打暗号或做手势，不准携带与考试课程内容相关的材料，不准携带具有发送、接收信息功能或存储有与考试课程内容相关材料的电子设备（如手机、智能手表、非教师允许的计算器等），不准抄袭或协助他人抄袭试题答案或者与考试课程内容相关的资料，不准窃取、索要、强拿、传、接或者交换试卷、答卷、草稿纸或其他物品，不准代替他人或让他人代替自己参加考试，等等。凡违反考试纪律或作弊者，按《北京大学本科考试工作与学习纪律管理规定》给予相应处分。

5、考生须确认填写的个人信息真实、准确，并承担信息填写错误带来的一切责任与后果。

诚信宣言：

我承诺，严格遵守校规校纪，诚信考试！

考生签名：_____

北京大学信息科学技术学院试题

考试科目： 数据结构与算法 A 姓名： _____ 学号： _____

考试时间： 2022 年 12 月 21 日 上午 任课教师： _____

一. 单选、多选或填空题（8 小题，每空 1 分，共 10 分）

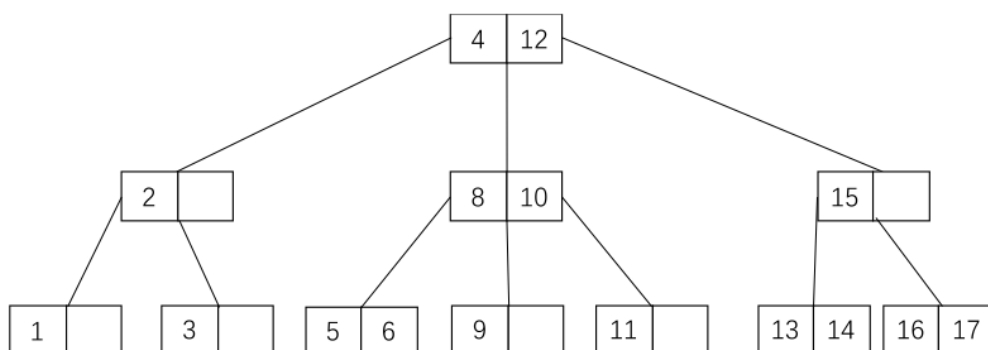
1. n 个顶点的连通的无向图用邻接矩阵表示时，该矩阵至少有（ ）个非零元素。
A. n B. $2(n-1)$ C. $n/2$ D. n^2
2. 无向图 $G=\langle V, E \rangle$ ，其中： $V=\{a, b, c, d, e, f\}$ ， $E=\{(a, b), (a, e), (a, c), (b, e), (c, f), (f, d), (e, d)\}$ ，对该图从 a 点开始进行深度优先遍历，得到的顶点序列正确的是_____。
A. a,b,e,c,d,f B. a,c,f,e,b,d C. a,e,b,c,f,d D. a,e,d,f,c,b
3. 对于以下几类排序算法，平均时间复杂度为 $O(n\log_2 n)$ 的有_____，属于稳定排序的算法有_____，在最好情况下时间复杂度可以达到线性时间的算法有_____。
A. 直接插入排序 B. 冒泡排序 C. 快速排序 D. Shell 排序 E. 归并排序 F. 选择排序
4. 针对序列 15, 9, 7, 8, 20, -1, 4 使用 Shell 排序算法进行排序，若第一趟排序结果为 15, -1, 4, 8, 20, 9, 7，则该趟排序采用的增量为_____。
5. 针对给定的关键码序列 110, 119, 763, 114, 120, 122 进行基数排序（按照最低位优先，关键码权重递增依次进行排序），则第 2 趟分配收集后得到的关键码序列为_____。
6. 使用置换选择排序对关键码序列 {10,7,9,4,8,5,6,3,2,1} 计算顺串，假设最小堆大小 $m=3$ ，最后可以得到_____个顺串，对应的关键码分别为_____。
7. 下列数据结构在动态增/删过程中，不能总保持树的高度为 $O(\log n)$ 有_____。
A. 二叉搜索树； B. 红黑树； C. AVL 树； D. 伸展树
8. 二维矩阵 $A[m][n]$ 以行优先方式存储，且每个元素占据 4 字节内存空间。已知 $A[0][0]$ 内存地址为 644， $A[2][2]$ 内存地址为 772，则 $A[3][3]$ 内存地址为_____。

二. 简答辨析题（6 小题，共 16 分）

1. 【3 分】有一个由点集 {0, 1, 2, 3, 4} 组成的完全无向图，其边的权值由以下矩阵给出：

$$W = \begin{pmatrix} 0 & 1 & 8 & 1 & 4 \\ 1 & 0 & 12 & 4 & 9 \\ 8 & 12 & 0 & 7 & 3 \\ 1 & 4 & 7 & 0 & 2 \\ 4 & 9 & 3 & 2 & 0 \end{pmatrix}$$

- 1) 用这个图生成一棵树，使得点 0 为树的叶子结点，求这棵树的最小权重并画出求解过程；
 - 2) 给出在该图中从点 1 到点 2 至多经过 3 条边的最短路径及其长度。
2. 【2 分】假设外存上有 120 个初始归并段，那么进行 12 路归并时，最少需要补充几个虚段才能实现最佳归并？（需要简要地给出推导过程）
 3. 【2 分】现有 8 个顺串，每个顺串的第一个记录的关键码分别为 11, 6, 20, 3, 9, 4, 20, 12, 第二个记录的关键码分别为 12, 15, 31, 7, 12, 5, 40, 13。画出对顺串开始 8 路归并时的败者树，以及输出一个优胜者后重构得到的败者树。顺串索引从 1 开始计数。
 4. 【2 分】将关键码序列 (16、4、23、2、17、9、14、11) 散列存储到散列表中。散列函数为： $H_1(x) = (x * 5) \text{ MOD } 7$ ，处理冲突采用双散列探查法，第二个散列函数为： $H_2(x) = 5 - (x \text{ MOD } 5)$ ，要求负载（装载）因子为 $8/11$ 。请画出所构造的散列表。
 5. 【4 分】如下是一棵三阶 B 树。假设
 - (1) B 树存储在一个磁盘文件中，并且每个结点都存在于一个独立的磁盘块中，一次磁盘的读写操作，会读写一个文件块；初始状态是 B 树在内存中没有缓存；
 - (2) 内存足够大，当插入数据时，在插入的检索阶段所读入的 B 树结点，缓存在内存中，后续需要读取该结点时候，不需要再从磁盘读入；
 - (3) 对 B-树的任何更新操作，需要保存到对应的磁盘文件中。



请回答下面的问题，每一道小题都需要简要说明理由。

- (1) 给出在上面的 B 树中，查找 6 的过程，并说明共进行多少次读盘；
 - (2) 给出在上面的 B 树中，查找 7 的过程，并说明共进行多少次读盘；
 - (3) 给出在上面的 B 树中，插入 7 的过程，并说明上述过程中有多少次对 B 树的访外操作；
 - (4) 对于 h 层的 B 树，一次插入操作最多读写次数是多少，给出推导过程。
6. 【3 分】Linux 内核中大量采用了红黑树作为存储结构，其理由是“最坏情况下，红黑树相比于 AVL 树与 Splay 树需要的旋转操作更少”。现在假设有一颗高度为 $h > 5$ 的红黑树，给出最坏情况下，插入操作所需的旋转次数以及原因。这里每次父子结点交换位置算 1 次旋转，即单旋转算 1 次，双旋转算 2 次。

三. 算法填空题（每空 1 分，共 5 分）

1. 下述算法实现在图 G 中计算从顶点 i 到顶点 j 之间长度为 len 的简单路径条数。图的 ADT 如下：

```
Class Graph {
public:
    int VerticesNum();
    int EdgesNum();
    Edge FirstEdge(int oneVertex);
    Edge NextEdge(Edge preEdge);
    bool IsEdge(Edge onEdge);
    int FromVertex(Edge oneEdge);
    int ToVertex(Edge oneEdge);
};

int visited[MAXSIZE]; // 初始化为 0
int GetPathNum_Len(Graph& G, int i, int j, int len) {
    if (____ /* 填空 1 */ ) return 1;
    sum = 0; // sum 表示通过本结点的路径数
    visited[i] = 1;
    for (Edge e = G.FirstEdge(i); G.IsEdge(e); e = G.NextEdge(e)) {
        int v = G.ToVertex(e);
        if (!visited[v])
            ____ /* 填空 2 */ ;
    } // for
    visited[i] = 0; // 本题允许曾经被访问过的结点出现在另一条路径中
    return sum;
} // GetPathNum_Len
```

2. 下面的代码实现了在线性时间内找到一个给定数组内从大到小第 k 个数。请将下面的代码补充完整。

```
int SelectK(int& a[], int l, int r, int index) { // 寻找序列 a[l ... r] 中第 k 大的数字
                                                // index = r+1-k

    int i = rand() % (r - l + 1) + l;
    swap(a[i], a[r]);
    int q = -1;
    int x = a[r], i = l - 1;
    for (int j = l; j < r; ++j)
        if (a[j] <= x)
            swap(a[++i], a[j]);
    swap(a[i + 1], a[r]);
    q = i + 1;
    if (____ /* 填空 3 */ ) {
        return a[q];
    }
    else {
```

```

        if ( _____ /* 填空 4 */ ) {
            return SelectK(a, q+1, r, index);
        }
        else{
            return _____ /* 填空 5 */ ;
        }
    }
}

```

四．设计分析题（2 小题，共 9 分）

注：本部分答题应申明算法思想，进行必要的算法复杂性分析，代码加以恰当注释。所给出算法的复杂度尽可能低；只需要给出伪代码或算法流程即可，不需要给出完整的代码。

1. 【5 分】给定一个有向图，结点数为 n ，分别标记为 $0, 1, \dots, n-1$ 。图中每条边为红色或者蓝色，边权都为 1，可能存在自环或平行边。使用邻接表存储（边表的每个元素额外记录该边为红或蓝色）。请设计一个算法，用尽可能低的复杂度求出结点 0 到所有其他结点的红色边和蓝色边交替出现的最短路径长度（如果不存在这样的路径，则长度为-1），并给出时间复杂度分析。

2. 【4 分】给定两棵 Splay 树 S 和 T ， S 和 T 中的关键码不重复，且 S 中的结点均小于 T 中的结点。

（1）试设计一个算法，实现函数 $\text{Merge}(S, T)$ 将它们合并为一棵 Splay 树。并分析时间复杂度；

（2）通过调用 $\text{Merge}(S, T)$ 操作，设计一个在 Splay 树 H 中删除结点 x 的函数 $\text{Delete}(H, x)$

五．期中成绩（30 分）

六．上机成绩（30 分）