|  |  |
| --- | --- |
| 得分 |  |

**1、填空题。**（每小题2分，本题满分2**0**分）

(1) 假设用一个一维数组B来按行存放一个对称矩阵A的下三角部分，那么访问A的第i行第j列元素的语句是：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_。(下标都从0开始)

(2) 在单链表指针P所指结点后插入指针为s的结点操作是： 。

(3) 线性表L=(a0,a1,a2,…an-1)用数组表示，假设删除表中任一元素的概率相同，则删除一个元素所需的平均移动次数为 。

(4) 在指针L指向一带头结点的双向循环链表，则判断该表中只有一个元素结点的条件

lLink data rLink

是： (设结点结构为 . )

(5）使用邻接矩阵表示图时，遍历一个顶点的所有相邻顶点的时间复杂度是：\_\_\_\_\_\_\_\_\_\_\_\_\_。(假设图的顶点个数为n, 边的个数为e)

(6) 已知带权有向图如下，使用Dijkstra算法求解从顶点1到达各顶点的最短距离时，该算法将按照\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_（列出顶点顺序）的顺序给出各个顶点的距离。

1 20 2 30 3 10 4

8 11

14 6 9 12 3

5 3 6 18 7 8 8

(7) 假设一组关键码为（20，41，22，17，19，56，35），则用堆排序的方法建立的初始最大堆为 。

(8) KMP模式匹配算法的时间复杂度是：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_。

(9) 设循环队列存放在数组A[0..m-1]中，若用牺牲一个单元的办法区分队列满和队列空(设队尾指针为rear, 队头指针为first)，则判断队满的条件是： 。.

(10) 对n个元素进行排序，如果用直接选择排序，所需的关键码比较次数最少为 　，如果用直接插入排序，则所需的关键码比较次数最少为 。

|  |  |
| --- | --- |
| 得分 |  |

**2、选择题。**（每小题2分，本题满分2**0**分）

（1）假设使用转换后的二叉树来表示森林，那么对该二叉树的前序遍历对应于对森林的\_\_\_\_\_\_遍历。

A 先根次序遍历 B 后根次序遍历 C 广度优先遍历 D 以上都不是

(2) 假设我们将一个表达式表示为一棵二叉树。比如a+b\*c对应的树的根为+，其左子树是一个叶子结点a，右子树的根为\*，左右子结点分别为b和c。显然每个内部结点代表了一个运算。假设一个并行CPU有很多个ALU可以同时执行计算任务，且完成所有计算需要的时间都是一个时间单位。那么完成一个表达式计算的最短时间是\_\_\_\_\_\_\_。

A 树的高度 B 树的宽度 C 树的内部结点个数 D 树的分支数

(3) 假设在快速排序算法中总是选择被排序子序列的最后一个元素作为基准。那么这个算法的最坏情况出现在\_\_\_\_\_\_\_。

A被排序的初始序列已经排好序时

B被排序的初始序列是逆序时

C被排序的初始序列呈现中间大，并逐次向两边减小的情况

D以上都不是

(4) 散列方法中，线性探查法的“堆积”问题是指\_\_\_\_\_\_\_。

A 不同探查序列的关键码占据了可利用的空桶，使得寻找某一关键码需要经历很长的探查序列。

B 不同探查序列的关键码占据了可利用的空桶，使得插入某个关键码时找不到空桶。

C 不同探查序列的关键码占据了可利用的空桶，导致寻找某个关键码时出现错误。

D不同探查序列的关键码占据了可利用的空桶，为了保证算法能够寻找到正确的关键码，不得不将装载因子限制在某个阀值之下。

(5) 在中序线索二叉树（使用lTag，rTag）中，一个结点的中序后继是\_\_\_\_\_\_\_\_\_\_。（多选）

A 如果其rTag为0，则后继是其rightChild所指结点的最左后代；

B 如果其rTag为0，则后继是其rightChild所指结点；

C 如果其rTag为1，则后继是其rightChild所指结点的最左后代；

D 如果其rTag为1，则后继是其rightChild所指结点。

(6) 假设有一棵二叉树的结点前序排列是1、2、3、4、5。下面的 序列不可能是这棵二叉树的中序排列。

A 3、2、1、4、5

B 2、3、1、4、5

C 2、1、4、5、3

D 2、1、5、3、4

(7) 有11个结点的AVL树的最大高度为\_\_\_\_\_\_\_。(假设叶结点的高度为1，树的高度为根结点的高度)

A 3 B 4 C 5 D 6

(8) 下面的排序算法中，时间复杂度不是O(nlog2n)的算法是\_\_\_\_\_\_\_。（多选）

A 折半插入排序 B堆排序 C 快速排序 D基数排序

(9) 使用Prim算法求解最小生成树时，使得算法效率较高的图的表示方式是\_\_\_\_\_\_\_\_\_。

A 邻接矩阵表示 B 邻接表表示 C 以上表示方法都一样

(10) 在B树的删除操作中，最坏情况下可能需要读写磁盘\_\_\_\_\_\_\_\_\_次。（假设内存工作区足够大，但操作之前各个结点都存放在磁盘上，B树的高度为h）

A 2h-2 B 2h-1 C 3h-1 D 3h-2次

|  |  |
| --- | --- |
| 得分 |  |

**３、解答题。**（每小题5分，本题满分20分）

(1) 使用一个32位整数以位向量集合的方式来表示0到31之间的整数的子集。请写出打印集合s中所有元素的代码。

(2) 假设一棵二叉树T的中序遍历序列和层次遍历（按层次递增顺序排列，同一层次自左向右）序列分别是ABCDGEF和CAEBDFG。请写出该二叉树的后序遍历序列。

(3) 已知图的邻接矩阵为:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | V1 | V2 | V3 | V4 | V5 | V6 |
| V1 | 0 | 1 | 1 | 0 | 0 | 0 |
| V2 | 0 | 0 | 1 | 1 | 0 | 0 |
| V3 | 0 | 0 | 0 | 0 | 1 | 0 |
| V4 | 0 | 0 | 1 | 0 | 0 | 1 |
| V5 | 0 | 0 | 0 | 0 | 0 | 0 |
| V6 | 0 | 0 | 0 | 0 | 1 | 0 |

请写出全部拓扑排序序列。

（4）对图3－1所示的AOE网络，求出每个活动的最早开始时间e()和最迟开始时间l(),并确定哪些活动是关键活动。



|  |  |
| --- | --- |
| 得分 |  |

**4、以下算法实现了循环链表的逆转。**

**template<class Type>**

**void** CList<**Type**>::Inverse ( ) {

CListNode<**Type**> \*p,\*q,\*r;

**if** (first->link= =first) **return**;

p=first->link;q=p->link;

p->link=first;

**while** ( )

{ r=q->link;q->link=p;

p=q;q=r;

}

;

}

（1）请在算法的横线上填入适当的语句, 每处只填一个语句或一个表达式；（6分）

（2）请用大“O”表示法表示算法的时间复杂度和空间复杂度，假设链表长度为n。（4分）

|  |  |
| --- | --- |
| 得分 |  |

**5、以下给出一个排序算法，其中n是数组A[ ]中元素总数。**

**template<class Type>**

**void** sort (**Type** A[ ]**, int** n) **{**

**int** i = 1**;**

**while** ( i<n-i+1)

{min=max=i;

**for** (j=i+1;j<=n-i+1;++j)

{ **if** (A[j]<A[min]) min=j;

**else if** (A[j]>A[max]) max=j;

}

**if** (min!=i) Swap(A[min], A[j]); // Swap()函数的功能是交换两个数

**if** (max!=n-i+1)

{ **if** (max= =i) Swap(A[min], A[n-i+1]);

**else** Swap(A[max], A[n-i+1]);

}

i++;

}

}

(1) 阅读此算法，说明此算法的基本思想。（2分）

(2) 对于下面给出的整数数组，请写出前4趟**while**结束时数组中数据的变化。（每行仅写出一个**while**循环的变化）（4分）

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 步 | A[0] | A[1] | A[2] | A[3] | A[4] | A[5] | A[6] | A[7] | A[8] |
|  | 34 | 23 | 89 | 26 | 3 | 45 | 18 | 102 | 41 |
| 1 |  |  |  |  |  |  |  |  |  |
| 2 |  |  |  |  |  |  |  |  |  |
| 3 |  |  |  |  |  |  |  |  |  |
| 4 |  |  |  |  |  |  |  |  |  |

（3）请写出此排序算法最好情况下数据的移动次数。(2分)

(4) 请用大“O”表示法表示算法的时间复杂度。（2分）

|  |  |
| --- | --- |
| 得分 |  |

**6、 程序设计题。**（每小题10分，本题满分20分）

(1) 二叉树T以二叉链表的形式存储，设计算法返回二叉树T的先序序列的最后一个结点的指针（要求不用递归）。

二叉链表的结点结构为：

leftchild data rightchild

(2) 已知一个有向图的邻接表表示，给出一个算法高效地计算出这个图的逆邻接表（即顶点i的边链表是所有进入该顶点的边）。要求算法的时间复杂度是O(n+e)，其中n是图的顶点数，e是图的边数。