

数据结构试卷

December 30, 2023

Dilettante258

wang1m@outlook.com

ABSTRACT

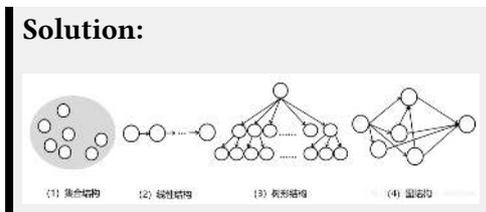
该 PDF 用 [Tpyst](#) 写作。上面的日期只是我写下这些内容的日期，不清楚这张试卷年份，用来记录我的学习过程同时方便后来人。

一、填空题(每小题 2 分，共 20 分)

1. 数据的四种基本逻辑结构

散列表的基本概念。

1. 数据的四种基本逻辑结构有树形结构、线性结构、_____结构和集合结构。



2. 散列表中， H 为散列函数， $H(42) = H(12)$ ，则 42,12 对于散列函数 H 而言称为_____。

Solution: 相关概念：

- 散列函数(h)：元素的关键字(key)与其存储位置(loc)之间的关系函数
- $Loc(key)$ ：表示关键字值为 key 的元素的存储地址。
- 散列表(hash, 哈希表)：用散列函数建立起来的表。
- 同义词：对同一散列函数，具有相同 h 值的关键字。
- 冲突： $key_1 \neq key_2$ ，但 $h(key_1) = h(key_2)$ 的现象。

3. 堆栈中 Top 指向顶元素，判断堆栈为空的条件是_____。

Solution:

```
BOOL IsEmpty(Stack *S)
{
    return S->top == -1;
}
```

4. 利用 AOE 网进行工程安排，完成工程所的最短时间是指从开始结点到完成结点的_____路径的长度，这条路径被称为关键路径。

Solution: “长度指各边的权值之和。补充：关键活动 - 关键路径上的活动，对整个工程的最短完成时间有影响，如果它不能按期完成就会影响整个工程。”

5. 若对某算法求得关键步骤执行次数 $f(n) = 20 + 30n \log_2 n + 40$ ，则其渐近时间复杂度为 $O(\quad)$ 。

2. 以下关键序列_____是一个最小堆。
 A.12,72,31,25,99,58 B.99,25,31,72,12,58 C.12,58,25,99,31,72 D.12,25,58,31,99,72

Solution: 先将序列画成完全二叉树形式,判断最小堆条件是否成立。双亲 \leq 子女

3. 后缀表达式: $8 \ 4 \ 4 \ + \ / \ 3 \ 3 \ * \ +$ 的值为_____。
 A.10 B.12 C.7 D.以上答案都不正确

Solution:

- ① 从左往右顺序扫描后缀表达式;
- ② 遇到操作数就进栈;
- ③ 遇到操作符就从栈中弹出两个操作数,并执行该操作符规定的运算;并将结果进栈;
- ④ 重复上述操作,直到表达式结束,弹出栈顶元素即为结果。

4. 对稀疏矩阵采用三元组方式存储的最主要目的是_____。
 A.使表达变得简单 B.使矩阵元素的存取变得简单
 C.去掉矩阵中的多余元素 D.压缩存储空间

Solution: 由于稀疏矩阵中只有少量非零元素,为了节省存储空间,对稀疏矩阵可以只存储非零元素。

5. 冒泡排序在好的情况下的时间复杂度是_____。
 A. $O(n)$ B. $O(\log_2 n)$
 C. $O(n^2)$ D. $O(n \log_2 n)$

Solution:

		时间复杂度			空间复杂度	稳定性	就地性	自适应性	基于比较
		最佳	平均	最差	最差				
遍历排序 $O(n^2)$	选择排序	$O(n^2)$	$O(n^2)$	$O(n^2)$	$O(1)$	非稳定	原地	非自适应	比较
	冒泡排序	$O(n)$	$O(n^2)$	$O(n^2)$	$O(1)$	稳定	原地	自适应	比较
	插入排序	$O(n)$	$O(n^2)$	$O(n^2)$	$O(1)$	稳定	原地	自适应	比较
分治排序 $O(n \log n)$	快速排序	$O(n \log n)$	$O(n \log n)$	$O(n^2)$	$O(\log n)$	非稳定	原地	自适应	比较
	归并排序	$O(n \log n)$	$O(n \log n)$	$O(n \log n)$	$O(n)$	稳定	非原地	非自适应	比较
	堆排序	$O(n \log n)$	$O(n \log n)$	$O(n \log n)$	$O(1)$	非稳定	原地	非自适应	比较
线性排序 $O(n)$	桶排序	$O(n+k)$	$O(n+k)$	$O(n^2)$	$O(n+k)$	稳定	非原地	自适应	非比较
	计数排序	$O(n+m)$	$O(n+m)$	$O(n+m)$	$O(n+m)$	稳定	非原地	非自适应	非比较
	基数排序	$O(nk)$	$O(nk)$	$O(nk)$	$O(n+b)$	稳定	非原地	非自适应	非比较

差 中 优

n 为数据量大小
 桶排序中, k 为桶数量
 计数排序中, m 为数据范围
 基数排序中, k 为最大位数, 数据为 b 进制

www.hello-algo.com

6. 已知一棵完全二叉树的第6层(设根为第1层)有8个叶结点,则该完全二叉树的结点格式最多是_____。
 A.52 B.111 C.119 D.127

Solution:

高度为 h 的二叉树恰好有 $2^h - 1$ 个结点时称为**满二叉树**。由此，高度为 h 的二叉树上至多有 $2^h - 1$ 个结点。

一棵二叉树中，只有最下面两层结点的度可以小于 2，并且最下一层的叶结点集中在靠左的若干位置上。这样的二叉树称为**完全二叉树**。（除最后两层外，其他层结点都是满度）

二叉树的第 $i (i \geq 1)$ 层上，至多有 2^{i-1} 个结点。完全二叉树结点个数 $2^{h-1} \leq n \leq 2^h - 1$ ，具有 n 个结点的完全二叉树的高度为 $\lceil \log_2(n+1) \rceil$ 。

第六层里有 8 个节点没有子树（也就是作为叶结点）， $sum=63+48=111$ 。

设这棵树最大层次为 7，前六层为满二叉树第 6 层(设根为第 1 层) \therefore 第六层里有 8 个叶结点(也就是没有子树) \therefore 第七层不满， $(2^6 - 1) + (2^{6-1} - 8) \times 2 = 63 + 48 = 111$

7. 对半搜索有序表(10,30,36,41,52,54,66,73,84,97)，在表中搜索关键字 34，则它依次与表中_____比较大小，最终搜索失败。
- A.52,41,30 B.54,84,66 C.52,30,36 D.54,73,66

Solution:

二分搜索有序表($a_0, a_1, a_2, \dots, a_{n-1}$)

- 如果有序表长 > 0 ，取表中某个元素 a_m 与 x 进行比较
- 如果 $a_m.key = x.key$ ，搜索成功，返回
- 如果 $a_m.key > x.key$ ，二分搜索有序表($a_0, a_1, a_2, \dots, a_{(low+high)/2-1}$)
- 如果 $a_m.key < x.key$ ，二分搜索有序表($a_{(low+high)/2+1}, a_{(low+high)/2+2}, \dots, a_{n-1}$)

8. 二叉搜索树中，关键字最大的结点_____。
- A.左子树一定为空 B.右子树一定为空
C.左右子树均为空 D.左右子树均不为空

Solution:

二叉搜索树：一棵二叉树，可以为空；如果不为空，满足以下性质：

1. 非空左子树的所有键值小于其根结点的键值。
2. 非空右子树的所有键值大于其根结点的键值。
3. 左、右子树都是二叉搜索树。

9. 如果线性表最常用的操作是读取第 i 个元素的值，则采用_____存储方式最节省时间。
- A.顺序表 B.带表头结点的单链表
C.不带表头结点的单链表 D.双向链表

Solution: 线性表中最常用的操作是取第 i 个元素，所以，应选择随机存取结构即顺序表，同时在顺序表中查找第 i 个元素的前趋也很方便。

单链表和单循环链表既不能实现随机存取，查找第 i 个元素的前趋也不方便，双链表虽然能快速查找第 i 个元素的前趋，但不能实现随机存取。

10. 设有向图 G 的边集为 $\langle 0,1 \rangle, \langle 1,2 \rangle, \langle 4,1 \rangle, \langle 4,5 \rangle, \langle 5,3 \rangle, \langle 2,3 \rangle$ ，则下面不属于该图的拓扑排序的是_____。
- A.041253 B.402153 C.045123 D.401523

Solution:

一个拓扑序列是 AOV 网中顶点的线性序列，使得对图中任意二个顶点 i 和 j ，若在图中， i 领先于 j ，则在线性序列中 i 是 j 的前驱结点。检测拓扑排序的方法 对图中每一条边 $\langle i, j \rangle$ ，在序列中 i 排在 j 之前。B 选项不符合 $\langle 1, 2 \rangle$ 。

三、简答题(每小题 8 分，共 48 分)

- 如图 1 所示顶点表示小区，边表示连接小区间的光纤，边上的权表示铺设纤需花费的造价。现在需要让小区间光纤通信畅通且总造价最省，请画出方案构建过程，并给出相应的总造价。

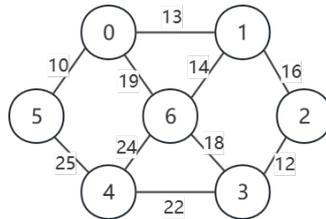


图 1

Solution:

构造网的一棵最小生成树，即：在 e 条带权的边中选取 $n-1$ 条边（不构成回路），使“权值之和”为最小。

普利姆算法 算法思想:

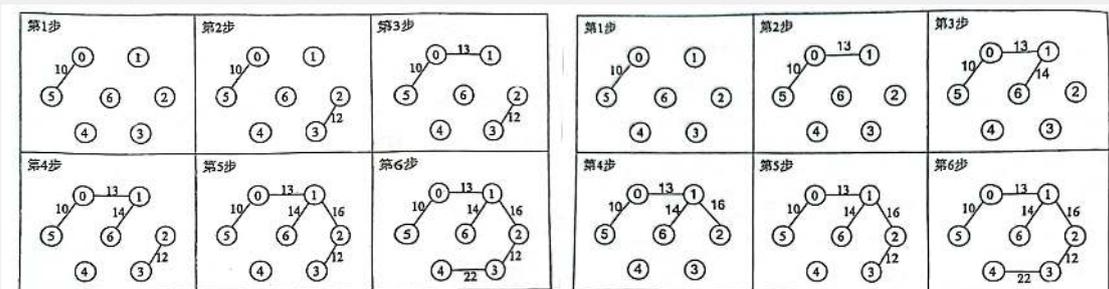
- 取图中任意一个顶点 v 作为生成树的根，之后往生成树上添加新的顶点 w 。
- 在添加的顶点 w 和已经在生成树上的顶点 v 之间必定存在一条边，并且该边的权值在所有连通顶点 v 和 w 之间的边中取值最小。
- 之后继续往生成树上添加顶点，直至生成树上含有 $n-1$ 条边为止。

克鲁斯卡尔算法 算法思想:

考虑问题的出发点: 为使生成树上边的权值之和达到最小，则应使生成树中每一条边的权值尽可能地小。

- 先构造一个只含 n 个顶点的子图 SG ，然后从权值最小的边开始，若它的添加不使 SG 中产生回路，则在 SG 上加上这条边
- 如此重复，直至加上 $n-1$ 条边为止。

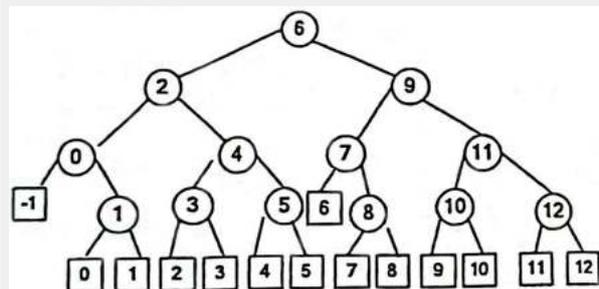
	普利姆算法	克鲁斯卡尔算法
时间复杂度	$O(n^2)$	$O(e \log e)$
适应范围	稠密图	稀疏图



两种过程。最小代价生成树的代价为 87。每步 1 分，代价计算正确得 2 分。

2. 对长度为 13 的有序表(假定下标从 0 开始标记)进行对半搜索, 请画出对应的二叉判定树。

Solution:



3. 请将图 2 所示的森林转成二叉树

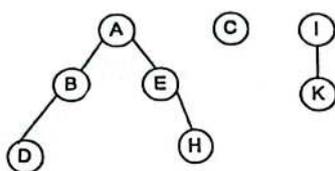
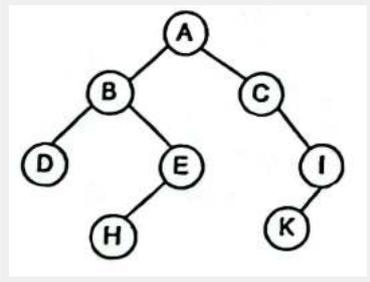


图 2

Solution:



4. 已知散列表如下所示, 长度 $M=11$, 依次输入关键字 17,18,37,74, 试以双散列法解决冲突, 散列函数为 $h_1(\text{key})=\text{key}\%11$, $h_2(\text{key})=\text{key}\%9+1$, 请填写散列表。

下标	0	1	2	3	4	5	6	7	8	9	10
元素						5	6		41		

Solution:

下标	0	1	2	3	4	5	6	7	8	9	10
元素	74				17	5	6		41		37

双散列法: 具备两个散列函数 h_1 和 h_2 , 探查序列为:

$h_1(\text{key}), (h_1(\text{key})+h_2(\text{key}))\%M, (h_1(\text{key})+2h_2(\text{key}))\%M, \dots$

例如, $h_1(\text{key})=37\%11=4$, $h_2(\text{key})=37\%9+1=2$, $(h_1(\text{key})+h_2(\text{key}))\%11=(4+2)\%11=6$ (重复!),

$(h_1(\text{key})+2h_2(\text{key}))\%11=(4+2*2)\%11=8$ (重复!), $(h_1(\text{key})+3h_2(\text{key}))\%11=(4+3*2)\%11=10$ 。

不打了, 打累了。

四、算法设计题（每小题 6 分，共 12 分）

二叉搜索树 T 用二叉链存结构表示，编写算法按递减顺序打印下中元素关键字的值。

相关结构体定义如下：

```
typedef struct elemtype {
    int Key;
    char Data;
} ElemType;
typedef struct bstnode {
    ElemType Element;
    struct bstnode *lchild;
    struct bstnode *rchild;
} BSTNode;
```

```
1 void ReduceOrder (BSTNode* p)
2 {
3     if(!p) return;           2分
4     else{
5         reduceOrder(p->rchild);    4分
6         printf("%d", p->Element.Key);
7         ReduceOrder (p->lchild);
8     }
9 }
```

C 语言

不带表头结点的单链表中元素各不相等，编写算法找出所有元素的最大值与链表第一个结点的值进行交换，如果成功，返回 1；否则，返回 0。

相关结构体定义如下：

```
typedef struct node{
    ElemType element;
    struct node *link;
} Node;
typedef struct singleList{
    struct node * first;
    int n;
} SingleList;
```

```
1 int Swap()
2 {
3     if(!first)
4         return 0;
5     Node *p, *max;
6     ElemType temp;
7     p = first-> link;
8     max = first;
9     while(p)
10    {
11
```

C 语言

```

    if(p->element > max->element)
12     max=p;
13     p=p->link;
14 }
15 if(max != p)
16 {
17     temp = first->element;
18     first->element = max->element;
19     max->element = first->element;
20 }
21 return 1;
22 }

```

填空题答案

题号	1	2	3	4	5	6	7	8	9	10
答案	图形	同义词	Top=-1	最长	$n \log_2 n$	n-m	160	有序	$n^2 - e$	F

单项选择题答案

题号	1	2	3	4	5	6	7	8	9	10
答案	B	D	A	D	A	B	C	B	A	B