考试形式 闭卷 考试时间 100 分钟

命题人 洋 林 2023 年 6 月 24 日 使用班级

教研室主任 年 月 日 教学院长 年 月 日

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 题号 | 一 | 二 | 三 | 四 | 五 | 六 | 七 | 八 | 总分 |
| 总分 | 60 | 40 |  |  |  |  |  |  | 100 |
| 得分 |  |  |  |  |  |  |  |  |  |

一、分析设计题（共4小题，每小题15分，共计60分）

1.（1）每个栈状态（1分），Min值（1分），共8分。

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Min: 5 |  | Min: 4 |  | Min: 2 |  | Min: 1 |
| 8 |  | 4 |  | 2 |  | 1 |
| 5 |  | 6 |  | 4 |  | 9 |
|  |  | 8 |  | 6 |  | 2 |
|  |  | 5 |  | 8 |  | 4 |
|  |  |  |  | 5 |  | 6 |
|  |  |  |  |  |  | 8 |
|  |  |  |  |  |  | 5 |

（2）参考Push操作：栈为空，m等于d，栈不为空时，m等于d与栈顶元素的m的较小值。（3分）

最终状态：（某个位置数值错误可以扣1分，最多扣4分）（4分）

|  |
| --- |
| d=1 | m=1 |
| d=9 | m=2 |
| d=2 | m=2 |
| d=4 | m=4 |
| d=6 | m=5 |
| d=8 | m=5 |
| d=5 | m=5 |

2.（1）F C K（字母、位置均正确得1分）（3分）

参考描述：缺少的字符为C F K。假设C为根，左子树结点为A、B，与先序矛盾。假设K为根，右子树结点为L、M、N，与先序矛盾。所以F为根，根据中序序列，判定C在左子树，K在右子树。（2分）

注意：描述时排除C、K为根即可。

（2）绘制二叉树（3分）

D

A

B

F

J

E

C

I

L

M

N

G

H

K

后序序列：A C B E D I H G K N M L J F（3分）

（3）参考描述（4分）：（可以结合图示说明，伪代码、代码均可成为描述手段）

设中序序列为S，先序序列为T，序列长度为n；

先序序列的首字母（设为R）为根，在中序序列中找到R，位置为m。

那么S的子序列SL（S1至Sm-1）为左子树的中序序列，SR（Sm+1至Sn）为右子树的中序序列。（SL或SR的长度可以为0）

同理T的子序列TL（T2至Tm）为左子树的先序序列，TR（Tm+1至Tn）为右子树的先序序列。（TL或TR的长度可以为0）

递归处理SL与TL，SR与TR即可还原二叉树。

S: S1 S2 ...... Sm-1 R Sm+1 ...... Sn

SL SR

T: R T2 ...... Tm-1 Tm Tm+1 ...... Tn

TL TR

3.（1）邻接矩阵与邻接表写成无向图的酌情判分。

邻接矩阵（3分）

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 1 | 0 | 3 | 3 | 6 | ∞ | ∞ | ∞ |
| 2 | ∞ | 0 | 4 | ∞ | 5 | ∞ | ∞ |
| 3 | ∞ | ∞ | 0 | ∞ | 4 | ∞ | ∞ |
| 4 | ∞ | ∞ | ∞ | 0 | ∞ | 5 | ∞ |
| 5 | ∞ | ∞ | ∞ | ∞ | 0 | ∞ | 3 |
| 6 | ∞ | ∞ | 3 | ∞ | ∞ | 0 | 7 |
| 7 | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ | 0 |

邻接表（3分）

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 |  |  | 2 | 3 |  |  | 3 | 3 |  |  | 4 | 6 |
| 2 |  |  | 3 | 4 |  |  | 5 | 5 | ^ |  |  |  |
| 3 |  |  | 5 | 4 | ^ |  |  |  |  |  |  |  |
| 4 |  |  | 6 | 5 | ^ |  |  |  |  |  |  |  |
| 5 |  |  | 7 | 3 | ^ |  |  |  |  |  |  |  |
| 6 |  |  | 3 | 3 |  |  | 7 | 7 | ^ |  |  |  |
| 7 | ^ |  |  |  |  |  |  |  |  |  |  |  |

（2）共6分，只写最终结果得3分。（明确边的选择次序即可）

2

5

3

1

6

4

7

2

5

3

1

6

4

7

2

5

3

1

6

4

7

3

3

4

5

3

3

前4步不分前后 第5步 第6步

如果使用语言描述求解过程的，酌情判分。

（3）多种参考答案：（3分）

深度优先搜素方式（简要）：使用栈存储路径，如果访问的结点在栈内，则出现回路。

广度优先搜索方式（简要）：访问结点是当前结点的前驱，则出现回路。

其他方式：酌情判分，主要思路是路径上结点出现重复即可。

4.（1）参考答案三选一。描述过程（3分）时间复杂度（2分）

选择排序：进行K轮选择操作，即可找到第K小的数。

参考序列变化：（如写出正确序列变化即可得分）

第一轮： **03** 12 35 78 63 42 37 19 08 24 55

第二轮： 03 **08** 35 78 63 42 37 19 12 24 55

第三轮： 03 08 **12** 78 63 42 37 35 19 24 55

第四轮： 03 08 12 **19** 78 63 42 37 35 24 55

时间复杂度：O(n2) 或 O(n\*K)

冒泡排序：进行K轮选择操作，即可找到第K小的数。

...... K轮冒泡

时间复杂度：O(n2) 或 O(n\*K)

插入排序：进行插入排序后，取出第K小的数。

...... n轮插入操作

时间复杂度：O(n2)

（2）两种参考答案。描述过程（3分），时间复杂度（2分）

二叉排序树：（如进行适当优化，酌情判分）

完成二叉排序树建树操作（画出最终二叉排序树即可得分），然后中序遍历至第K个数即可。

时间复杂度：O(n\*logn)

堆（小顶堆）：（如进行适当优化，酌情判分）

将n个数依次进堆（建堆操作），第K次取堆顶元素即可。

时间复杂度：O(n\*logn）

（3）描述过程（3分），时间复杂度（2分）。（其他算法酌情判分）

参考答案：应用快速排序方式，取参考值，对序列进行比较与交换。一轮快速排序后，序列分成两部分，如下所示：

S1 S2 ...... Sm | Sm+1 ...... Sn

如果K等于m，则Sm即为结果；

如果K小于m，在S1至Sm中递归寻找第K小元素；

如果K大于m，在Sm+1至Sn中递归寻找第K－m小元素。

时间复杂度：O（logn \* logn）或 O( (logn) 2 )

二、综合应用题（共2小题，每小题20分，共计40分）

5.（1）参考回答：（3分）

两个结点拥有相同的根（或者拥有相同得前驱结点）

（2）绘制树或森林（3分）

1

5

3

2

7

8

6

9

4

5与9是朋友关系，他们有共同的前驱（在同一棵树上）。（1分）

4与7不是朋友关系，他们没有相同的前驱（不在同一棵树上）。（1分）

（3）Find(x)参考答案（2分），Union(x,y)参考答案（2分）

Find(x)描述：

如果x的前驱不等于x，将前驱赋值给x，直到x的前驱等于自己（根）

Find(x)伪代码：

while(x前驱不等于x)

x = x的前驱

返回x

Find(x)代码：

int F[];//前驱数组

int Find(int x){

while(x != F[x])

x = F[x];

return x;

}

Union(x,y)描述：

x的根如果不等于y的根，将y（或x）的前驱设置为x（或y）。

Union(x,y)伪代码：

if(x的根 != y的根)

y的前驱 = x // 或者 x的前驱 = y

Union(x,y)代码：

int F[]; //前驱数组

void Union(int x, int y){

x = Find(x);

y = Find(y);

F[y] = x; //或 F[x] = y;

}

（4）**答案不唯一。**填空（3分），绘制树或森林（3分）

判分标准：

1，2，5，7在同一棵树上，3，4，6，8，9在同一棵树上。

注意：前驱值与树必须正确对应。

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| parent | 7 | 1 | 4 | 8 | 2 | 4 | 7 | 8 | 8 |
|  | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |

7

2

1

8

3

5

9

6

4

（5）判分依据：树的个数等价于“根”的个数。（2分）

参考答案：结点的前驱值等于自身的结点的个数。

（7的前驱为7，7为根，8的前驱为8，8为根。所以树的个数为2）

6.（1）参考描述（3分），时间复杂度（2分）

找出距离为K（空旷指数）的两个相邻路标Si与Si+1，将路标放置在两个路标的中间处。（中间处：（Si+Si+1）/ 2，向上取整、向下取整均可）

时间复杂度：O(n)。

（2）参考算法：参照法（反证法）（5分）

寻找所有间距大于K的路段，在该路段中每间隔K处放置一个路标，使得间距最大为K。如放置路标的个数小于等于M，则“空旷指数”可以达到K，反之不可。

（3）参考证明：（方法较多）（5分）

使用贪心策略，不断寻找最大间距，将路标放置其中心处。即可使得“空旷指数”不增加。

（4）参考答案：二分查找（答案）

设路段S的原始“空旷指数”为K。在[1,K]中折半取值m为新空旷指数进行尝试。

如新“空旷指数”所需路标小于等于M（还可能缩小），则在[1,m]范围继续尝试。

如新“空旷指数”所需路标数量大于M（必须扩大），则在[m+1,K]范围继续尝试。