考试形式 闭卷 考试时间 100 分钟

命题人 肖 程 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

姓 名 班 级 　 学 号

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 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.设计一个栈，使它可以在O(1)的时间复杂度内实现Push（进栈）、Pop（出栈）、Top和Min操作（Min操作是指得到栈中元素的最小值）。

（1）如果将5，3，8，6，7，4，2，9，1依次进栈，操作的序列为Push、Push、Min、Pop、Push、Min、Push、Push、Pop、Push、Min、Push、Min、Push、Push、Min。第一次Min操作时，栈的形态如图1所示，通过遍历栈中元素得到最小值为3。请在图1右侧并仿照图1画出后续每次Min操作时栈的形态并写出Min操作结果。

Min: 3

3

5

图1

（2）设计Push操作算法，同时将d（数据）与m（最小值）进栈，通过获取栈顶元素的m值，可以得到当前栈中数据的最小值，实现了Min操作的复杂度为O(1)。请描述Push操作过程并画出栈的最终形态，如图2样例所示。

d=3 | m=?

d=5 | m=?

图2

2.给出二叉树的中序序列与被“遮盖”的先序序列。

中序序列：A B C D E F G H I J K L M N

先序序列：█ D B A █ E J G H I L █ M N

（1）按顺序写出被遮盖的结点，并描述依据。

（2）绘制出这颗二叉树并写出后序序列。

（3）通过给定的完整中序序列与先序序列，设计并描述根据两组序列还原二叉树的算法。

3.已知有向图G如图3所示。

2

5

3

1

6

4

7

3

4

5

3

7

3

4

3

6

5

图3

（1）写出图G的邻接矩阵与邻接表。

（2）若将该图视为无向图，使用Kruskal算法，求最小生成树。（使用文字描述、表格、图示均可）

（3）设计并描述算法，判断有向图是否存在回路。

4.设计算法找出长度为n的序列S中第K小的数。现给出样例如下：

K：4

S：12 08 35 78 63 42 37 19 03 24 55

（1）使用选择排序、冒泡排序或插入排序任意一种排序算法实现目标，通过样例描述其过程及时间复杂度。

（2）使用二叉排序树或堆结构实现目标，通过样例描述其过程及时间复杂度。

（3）借鉴快速排序算法流程，设计尽可能高效的算法实现目标，通过样例描述其过程及时间复杂度。

二、综合应用题（共2小题，每题20分，共40分）

5.已知部分学生之间的朋友关系，设计算法确定任意两人是否为朋友关系。

规定：x 和 y 是朋友，y和z是朋友，那么x和z也是朋友。如果x，y是朋友，那么x的朋友都是y的朋友，y的朋友也都是x的朋友。现使用树形结构表示朋友关系，即一棵树T上所有的结点均为朋友关系。

（1）描述判断两个结点在同一棵上的方法。

（2）树T选择使用顺序表存储前驱结点值。初始时每个结点的前驱均为自身，即每个结点都是一棵树的根。图4表示了一个朋友关系样例，画出样例所描述的朋友关系对应树或森林，并分别说明5与9、4与7之间是否为朋友关系关系并说明理由。

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

图4

（3）现设计以下操作：

Find(x)：查找结点x的根。

Union(x,y)：合并x结点与y结点的所在的树。

分别描述Find与Union的算法流程（使用语言描述、图文解释、伪代码、C与C++代码均可）

（4）从初始状态依次执行Union(1,2)、Union(4,3)、Union(2,5)、Union(4,6)、Union(7,2)、Union(8,9)、Union(9,6)，将最终的前驱值填入图5，并画出最终状态的树或森林。

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

图5

（5）设计并描述求朋友分组个数（树的个数）的算法。

6.在公路的某些位置设置路标，以降低“空旷指数”。相邻路标的最大距离K定义为该公路的“空旷指数”。路段使用n个正整数的升序序列S表示，S1为起点，Sn为终点，序列值Si代表路标放置位置。显然起点与终点均有路标，且路标位置必须为整数。

样例路段S：1 2 7 9，如图5所示：

9

7

2

1

图5

（1）已知路段S，如果再提供1个路标，描述其最优的放置位置以尽可能的降低“空旷指数”K，并说明其时间复杂度。

（2）已知路段S，如果再提供M个路标，空旷指数能否达到K？设计并描述该判定算法并说明其时间复杂度。

（3）已知路段S，空旷指数为K，再额外提供M个路标，证明M越大，“空旷指数”的值只会不变或变小，且在[1，K]之间。

（4）已知路段S，再额外提供M个路标，设计高效算法求出最小“空旷指数”。描述其过程及时间复杂度。