EDA软件设计 I Quiz 2

学号（0.5分）：

姓名（0.5分）：

1.（4’）从“好”到“坏”的顺序，列出下列算法复杂度的排序 （Note: ^代表幂运算，n^(3/2)是n的3/2次幂）

O（n^(3/2)) 、O(n) 、O(n^(3/2) + n)、O(logn)、O（n\*logn）

O(logn) 、O(n)、O(n\* logn) 、{O(n^(3/2))、O(n^(3/2) + n)}

{ }: 等价复杂度

1. (3’) BFS时间复杂度和空间复杂度是？

时间：O(V+E)

空间：O(V)

1. (2’) 在大规模图上，通常可以如何改进 BFS 来解决s-t 问题（寻找已知起始点和终止点之间路径问题）——提示：一个现有算法

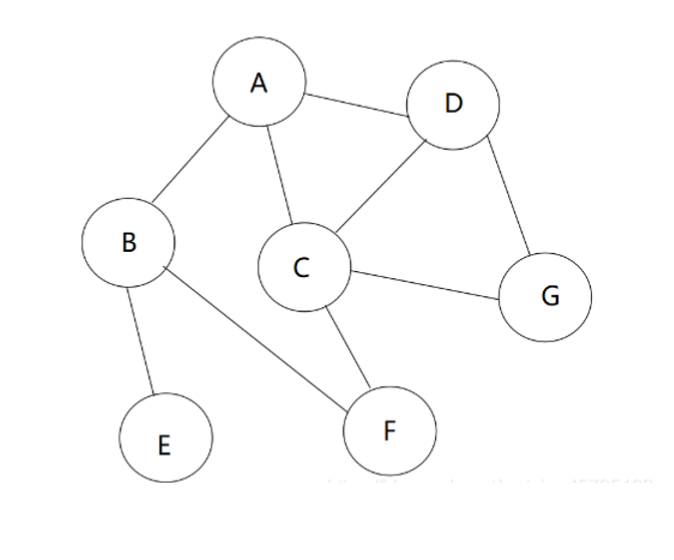
Bidirectional BFS （双向 BFS）

1. (3’) 常见的算法设计策略（e.g, 分治法），除分治法再写三个

动态规划、贪心、回溯

1. (4’) DFS 的两种实现方式是： 栈、递归
2. (5’) 从 F点开始，run DFS，选择下一个节点时如有多个选项，选择字母顺序表中靠前的那个

F-B-A-C-D-G-E



1. (3’) 写出天然适合递归解决的三类问题

树结构、分治法问题、具有嵌套结构的问题、一些具有递归性质的数学问题

1. (2’) 递归的基本结构是由哪两部分构成的

基准情况、递归调用

1. (2’) “能使用显示栈（Stack）来实现的算法，都可以用递归来实现”——这个陈述对吗

正确，任何使用显式栈（如手动管理的栈数据结构）实现的算法，都可以通过递归来实现。

**原因如下：**

**等价性**：递归和使用显式栈的方法在本质上是等价的。递归调用函数时，系统会自动使用调用栈来保存函数的状态和局部变量。这与手动使用栈来保存状态的方式功能相同。

**状态保存**：在递归中，每次函数调用都会自动保存当前的执行状态。这相当于将状态压入栈中。当函数返回时，状态被弹出栈。这与显式使用栈来管理状态的过程一致。

**可转换性**：许多使用栈的算法，如深度优先搜索，都有对应的递归实现。实际上，任何基于栈的迭代算法都可以转换为递归形式，反之亦然。

**需要注意的是：**

**性能和资源限制**：虽然理论上可以互相转换，但在实际应用中，递归可能会导致调用栈过深，导致栈溢出。因此，在实现时需要注意可能的资源限制。

**尾递归优化**：某些语言支持尾递归优化，可以在一定程度上缓解递归带来的栈空间问题。

1. (3’) 有向无环图的三种应用： 课程排序（课程表）、程序构建、任务调度和依赖管理、版本控制系统、解析嵌套表达式、电路设计与分析、数据处理与工作流管理、区块链……

1. (2’) 两种拓扑排序算法分别是什么： kahn算法（卡恩算法）、基于 DFS的拓扑排序算法
2. (5’)写出下图中一种拓扑排序的结果： {5,7,3}, {1,0}, {2,6,4}

Note: { }内的顺序可完全打乱，但{ }群组之间的顺序必须确定如上，比如{7,3,5}, {0,1}, {6,2,4}

{3,7,5}, {1,0}, {2,4,6}……

