数据结构 – 上机考试

学生信息管理

考试时间：90 分钟 · **语言：**Java 或 Go（任选） · **计算机：**机房的计算机
允许使用：本课程的示例程序和幻灯片、自己的笔记 · **不允许使用：**互联网、人工智能（AI）工具、其他同学的帮助

任务

我们要管理一所大学的学生信息。每个学生有一个**学号**（`id`，整数）、**姓名**和**分数**（0–100）。请用**你自己实现的数据结构**完成初始代码中标有 **TODO** 的操作！

	Java	Go
初始代码	`practical/java/StudentRegistry.java`	`practical/go/main.go`
运行	`java StudentRegistry.java`	`go run .`（在 `go` 文件夹中）
解答中禁止使用	`java.util` 中的集合类（`ArrayList`、`LinkedList`、`HashMap`、`TreeMap` …）	`container/list`、`map`、内置排序

程序最后会**按级别**输出哪些检查通过（**OK**），哪些失败（**FAIL**）。**不要修改**检查部分！程序的输出是英文的。

各个级别

必须**按顺序**完成各个级别：只有前面所有级别都完全正确时，一个级别才算数。

第 1 级 – 及格（2）：单链表

- `add(s)` – 新学生加到链表的**末尾**
- `size()` – 学生人数
- `find(id)` – 按学号查找（找不到：`null` / `nil`）
- `toString()` / `String()` – 例如 `[1003 Li Wei 87, 1001 Wang Fang 92]`，空链表：`[]`

第 2 级 – 中等（3）

- `remove(id)` – 删除学生；如果有这个学生则返回 true（注意删除**第一个**结点的情况！）
- `averageScore()` – 分数的平均值（空链表为 0）
- `best()` – 分数最高的学生；并列时取链表中靠前的

第 3 级 – 良好（4）

- `addSortedByScore(s)` – 插入后链表仍按分数**降序**排列；分数相同时，新学生排在已有学生的**后面**（稳定插入）
- `reverse()` – **原地**逆置链表，不创建新结点

第 4 级 – 优秀（5）：二叉搜索树（键：学号）

- `insert(s)` – 插入；重复的学号**不**再插入
- `find(id)` – 按学号查找
- `height()` – 树的高度（空树：−1，只有一个结点：0）
- `inorderIds()` – 按中序排列的学号，例如 `[1001, 1002, 1003]`

第 4 级的检查还会进行一次**测量**（在链表和二叉搜索树中各查找 20 000 次）。请观察结果 – 口试时会问到！

评分

1. **自动检查：**程序输出的级别是**最高可能**的成绩。
2. **口试**（约 5 分钟）：你必须能够讲解自己的代码，并回答 2–3 个问题（例如 `find` *操作需要多少步？删除第一个结点时会发生什么？*）。
   如果讲解不能令人信服，成绩**可能会降低**。
3. 如果程序不能编译、使用了禁止的集合类，或者修改了检查部分，解答**不予接受**。

提示

- **逐级**进行，每写完一个方法就运行一次程序！
- 考虑**特殊情况**：空链表、只有一个结点的链表、第一个和最后一个结点。
- 第 9、10、14、15 周的示例程序很有帮助。
- 经常保存你的工作！

姓名：..... **学号：**..... **所选语言：**Java / Go