上机考试 – 评分指南

准备工作

- 复制到学生计算机上的内容：`practical/java/` 和 `practical/go/` 文件夹（初始代码），以及本课程的幻灯片和 `code/` 示例程序。
- **不得**放到学生计算机上的内容：`practical/solution/` 文件夹（Java 和 Go 的参考答案）。
- 在计算机上检查：`java -version`（建议 21，至少 17）和 `go version`（1.23+）。尽可能禁止访问互联网。
- 初始代码由参考答案生成：`python3 eszkozok/kiindulo_kod.py <答案> <输出> java|go` – 这样可以保证检查部分完全相同。

评分过程（每个学生约 5–7 分钟）

1. **运行：** `java StudentRegistry.java` 或 `go run .` – 程序输出自动检查达到的级别。
2. **快速检查代码：**
   - 是否使用了禁止的集合类（`ArrayList`、`LinkedList`、`HashMap`、`container/list`、`map`...）？
   - 是否修改了检查部分（`main` 和 `check` 函数）？
   - 是否“作弊”（例如 `toString` 返回固定的文本，或者 `find` 不是在自己的链表中查找）？
3. **口试问题：**根据学生的级别，从下面的问题中选 2–3 个。
4. **成绩：**自动检查的级别；如果学生不能讲解自己的代码，**可以降低一级**；严重时（明显不是自己完成的）判为不及格。

口试问题

第 1–2 级

- `add` 操作需要多少步？怎样才能做到 O(1)？*(End 指针 / tail)*
- 在 `remove` 中删除第一个结点时会发生什么？为什么还要记住前一个结点？
- 链表为空时 `find` 返回什么？程序为什么不会崩溃？
- 为什么在链表中 `find` 是 O(n)?为什么不能用二分查找？

第 3 级

- 你是怎样让 `addSortedByScore` 保持稳定的？ *(条件中用 `>=` – 越过相等的元素)*
- 在一个有三个结点的链表上演示 `reverse` 的过程！为什么要保存 `next` 指针？
- 有序插入需要多少步？*(O(n))*

第 4 级

- 测量结果说明了什么？为什么二叉搜索树更快？ *(平均 O(log n) 与 O(n))*
- 如果学号按**递增**顺序插入，树的高度是多少？这会导致什么？
- 为什么中序遍历给出有序的序列？
- 怎样删除有两个孩子的结点？ *(中序后继)*

需要注意的常见错误

错误	在检查中的表现
`add` 没有处理空链表	第 1 级出现 `NullPointerException` / nil 指针 panic
`remove` 删除第一个结点时没有修改表头	“deleting the first element” FAIL
`best()` 由于用了 `>=` 而返回后面的相等元素	“Best() -> Wang Fang” FAIL
`addSortedByScore` 用了 `>` 条件 – 不稳定	“stable order” FAIL
`reverse` 创建了新结点	检查能通过 – 要在**口试**中提问！
二叉搜索树没有处理重复的学号	“a duplicate id is not inserted” FAIL

另一种方式：作为课外大作业

这个任务也可以作为**课外大作业**布置（和以前的课程一样）：学生在第 16 周之前一周拿到题目，在第 16 周进行**口头答辩**。这时口试所占的比重更大（10 分钟，至少 4 个问题），学生必须能够讲解解答的每一行。