

# Debug Lab 2

## Intro

助教最近写二叉树写烦了，于是决定写一个多叉树。助教把这个多叉树的实现放在了 `tree.h` 里面。但是他对自己的代码不是很自信，所以他想要你帮他写一些测试代码，来验证他的代码的正确性。如果你测出了问题，请你帮助助教修复他的代码。

## 要求

本实验提供一个头文件 `tree.h`，其中实现了多叉树类型 `tree_t`，支持以下操作：

### 1. 插入一个键/值对

```
void tree_t::set(int key, int value);
```

当这个函数被调用时，`tree_t` 会在树中插入一个键/值对，如果这个键已经存在，那么就会覆盖原来的值。

### 2. 删除一个键/值对

```
void tree_t::remove(int key);
```

当这个函数被调用时，`tree_t` 会在树中删除一个键/值对，如果这个键不存在，那么什么也不会发生。

### 3. 根据键查找对应的值

```
int tree_t::get(int key);
```

当这个函数被调用时，`tree_t` 会在树中查找键 `key` 对应的值，如果这个键不存在，那么会引发一个异常。

该文件需要编译器以 C++ 17 或以上版本编译。推荐使用 CMake 进行构建。我们提供了一个 `CMakeLists.txt`，它将设置 CMake 项目，自动检测编译器版本并设置合适的编译选项。

`tree.h` 中可能包含的 BUG 有：逻辑错误，运行时错误，`use-after-free`，内存泄漏。你的任务是找出尽可能多的 BUG 并修复之。当然你不能引入新的 BUG。

---

**hint:** 分发代码中的 `main.cpp` 并不会触发所有的 BUG，你需要自己编写单元测试代码来对头文件进行全面的测试。在测试和调试过程中你可以借助各种工具，例如 GDB、ASAN、Fuzz Testing 等等。

同时需要注意的是，即使程序在你的电脑上能正常执行，也不代表它是正确的。一个没有 BUG 的程序应该在任何环境的任何输入下都能正常执行。

## 提交代码

你需要提交一个 7z 压缩包，包含两个文件：

- `tree.h`：修复后的头文件
- `report.pdf`：实验报告，包含以下内容：
  - 你是如何测试 `tree.h` 的

- 你发现的 BUG 的类别
- 你是如何发现这些 BUG 的
- 你是如何修复这些 BUG 的

这两个文件应当放在压缩包的根目录下。**请不要在压缩包内创建任何子目录。**

请注意你在修复 BUG 的过程中，可以任意地修改 `tree.h` 和 `main.cpp`。然而你最终需要清理代码，并**只提交** `tree.h`。在对代码进行评测时我们会使用一个不公开的 `main.cpp` 进行编译。

为了防止你通过重写代码来解决问题，我们会首先使用 `clang-format` 工具将代码宽度限制到 80 个字符，然后使用 `Unix diff` 工具来比较你上传的 `tree.h` 和原始的 `tree.h`。如果你的代码和原始代码相差超过 **20 行**，那么评测将拒绝进行。

在线评测将只会检查编译问题和 `diff` 行数。在实验截止提交后，我们将会对你的代码进行充分的评测，以检查你的代码是否真的修复了 BUG。我们的评测分为多个部分，如果你只修复了一部分 BUG，也可能得到部分分数。

Linux/Mac 用户可以使用下面的命令来检查 `diff`，这也是评测时用的命令：

```
diff tree.h tree-orig.h | grep -cE '^[<>]'
```

Windows 用户可以使用 PowerShell 命令来检查 `diff`：

```
(Compare-Object (Get-Content ./tree.h) (Get-Content ./tree-orig.h)).Length
```