

# Homework 4: Splay Tree V.S. AVL Tree

---

Splay 树是对二叉搜索树改进后的数据结构，它基于这样一种观点：在一些场景中，程序的数据访问具有良好的局部性，刚刚访问过的数据及其附近的数据有很大概率会被再次访问。因此 Splay 树会通过树节点的旋转操作将刚刚访问过的数据提到根节点的位置，形成一个类似缓存的机制。

本次实验的目的是对比 Splay 树和 AVL 树的数据访问开销。

## Part 1: 实验步骤

---

1. 实现一个 Splay Tree 和 AVL Tree;

*Tips 1: 为了减少同学们的工作量，允许同学们从网上寻找 Splay Tree 和 AVL Tree 的源码实现，如：*

[Splay Tree](#)

*Tips 2: 请在提交的作业中标明所参考的来源*

2. 生成数据量为  $n$  的输入集  $N$ ，选取至少 5 组不同的  $k/n$  值（尽量分散，同时要有  $k/n$  较小的情况，例如 5%, 25%, 45%, 65%, 85% 五组），并从  $N$  中选取连续  $k$  个数据作为查找集以保证查找集数据的局部性；
3. 分别对 Splay Tree 和 AVL Tree 进行测试，采用相同的输入集  $N$  插入数据，分别测试每组  $k/n$  值情况下，进行  $m$  次查找所花费的时间开销，并将实验的结果记录下来；
4. 分析实验数据，简单分析在 Splay Tree 是否体现出相较于 AVL Tree 的性能优势（可以从  $k/n$  值的选取、应用场景的角度展开分析）

*Tips 3: 实验中的参数的选取比较重要，包括  $m$ 、 $n$ 、 $k$ ，三者的含义如下：*

$m$ ：查询节点的次数，至少大于  $n$  两个数量级；

$n$ ：树中的节点总数；

$k$ ：查询的数据集的大小；

通过选取合适的参数值  $m$ 、 $n$ 、 $k$ ，满足  $m \gg n \gg k$  时，Splay Tree 相较于 AVL Tree 会有较明显的优势；

## Part 2: 提交要求

---

你提交的内容应该包括：

- 你的测试参数的选取情况以及测试数据集；
- 你的实验结果数据以及对实验结果的简单分析；
- Splay Tree 和 AVL Tree 源码的来源；

## Part 3: 注意事项

---

- 请将相关的代码和实验报告打包上传 Canvas，命名使用“学号+姓名+hw4”，如“521123456789+张三+hw4.zip”。
- **请勿抄袭！** 课后作业采用倒扣分制，如果有遗漏或者得分不足会在最终成绩酌情减分，同时课后作业的内容会体现在期末试卷中，对同学们也是一种练习。

- 本次作业的截止时间是 **2023年3月22日23:59**，迟交将会酌情扣分。
- 有任何作业相关的问题可以询问 江珣璠、熊天磊 助教。