

Splay 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。Splay 树虽然没有像 AVL 树那样达到尽可能小的树高，即不是对所有数据都可以保证一个相对小的访问代价，但是对频繁访问的数据提升很大。

本次实验的目的是对比 Splay 树和 AVL 树的数据访问开销。为减少同学们的工作量，本次作业允许同学们从网上寻找 Splay、AVL 树源码，如：[Splay-Tree](#)

实验需要首先对 Splay 树插入数据，并从插入的数据中寻找一个数据子集作为查找集，对查找集中的数据随机进行一定次数的访问。设插入的数据量为 n ，要对数据结构查找次数为 m ，其中查找集的数据量为 k 。在 Splay 树的设计中，当 k 远小于 n 远小于 m 时，Splay 树相比 AVL 树会有较明显的优势。

实验需要你：

1. 选定一个较大的 m 值：至少大于 n 两个数量级
2. 生成数据量为 n 的输入集，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，如 5%, 25%, 45%, 65%, 85% 五组），并选取连续 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 树和 AVL 树进行测试，选用相同的输入集进行数据插入。
分别测试每组查找集下查找 m 次所花费的时间。将 Splay 树和 AVL 树在不同的查找集下查找 m 次所花费的时间记录在表格中。
4. 分析数据并讨论 Splay 树在该实验场景下是否比 AVL 树有优势，性能优

势和 k/n 的关系，分析什么场景下使用 Splay 树有优势。

该实验结果需要写成一个报告，报告中需要说明：

1. 你选取的输入集以及查找的次数 m
2. 你选取的至少五组不同 k/n 的查找集
3. 你的测试方法，测试结果
4. 对实验结果的分析以及对 Splay 树的理解