

## Scapegoat Tree 性能分析书面报告

1. 选取数据范围为 0-20000，数据规模为 20000 的数据集，进行插入、查找、删除操作。

选取 $\alpha$ 值分别为 0.6、0.65、0.7、0.75、0.8，实验得时间如下：

$\alpha$	0.6	0.65	0.7	0.75	0.8
插入时间/ms	29.36	17.51	15.21	14.74	12.99
查找时间/ms	3.79	3.96	4.05	4.23	4.33
删除时间/ms	207.01	229.77	310.06	372.05	451.22

符合理论（ $\alpha$  越大，对应的树平衡要求越低，插入所需的平衡开销更小，查找和删除等操作会更慢）。

2. 在相同的数据集下，对 AVL 树进行插入、查找、删除操作，实验时间如下：

	AVL 树
插入时间/ms	42.27
查找时间/ms	1.47
删除时间/ms	1.48

由数据可见，像比于 Scapegoat Tree，AVL 树插入性能较差，查找和删除性能较强。这是因为 Scapegoat Tree 不强制保证所有子树完全平衡，允许一定程度的不平衡存在，因此在插入时用于维护树平衡的时间更少，但是相应的树高更高，且每次删除重构需要重构整棵树，导致查找和删除操作开销较大。

3. 由上述实验得，两者相比，AVL 树更适用于插入数据集较小，但查找和删除频繁的场景；Scapegoat Tree 更适用于插入数据集较大，查找和删除次数较少的场景，并且可以根据需求调整 $\alpha$ 大小：如插入数据较多、查找较少时，适当增大 $\alpha$ 的值；如插入数据较少、查找较多时，适当减小 $\alpha$ 的值