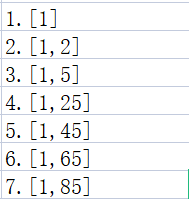
**HW5：AVL树与Splay树查找性能比较 实验报告**

本次实验主要目的是通过自选查找区间的方式，来检测在进行大量的局部查找时avl树与splay树之间的性能差异。主要实验过程如下：

1. 选择测试区间：

插入集合：[1, 100];因此n取值为100；

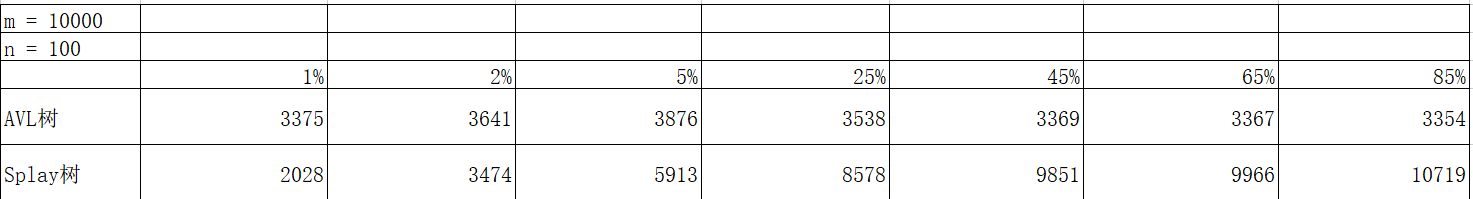
查找区间：

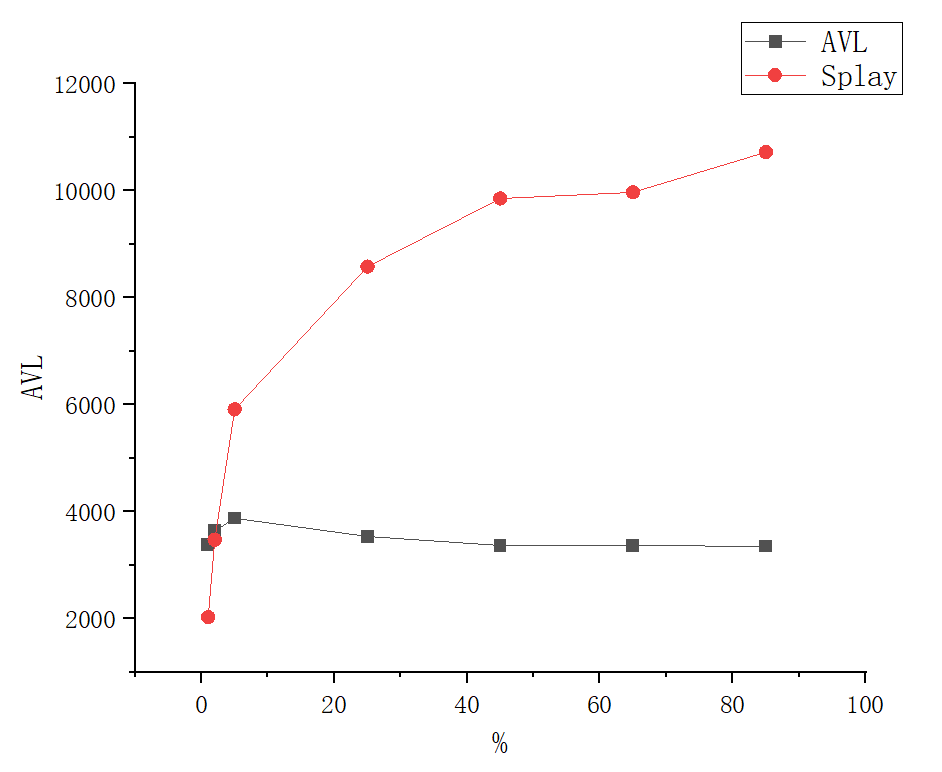


M的取值：10000次。

1. 测试方法：利用循环，通过对循环变量进行取余的方式生成对应的k的测试集，进行对应数据的查找，同时统计时间。
2. 测试过程：按照上述方式开展测试，期间通过多次实验取平均值、改变对应的测试集（在原来的测试集上面加一定的偏移量）等方式来生成对应的结果。结果图表如下：

（时间均以 微秒 为单位，使用Windows提供的QueryPerformanceCounter进行操作）





数据对应的折线图如上图所示。从图上可以看出，当取得百分比小于5%时，splay的查找性能比较明显的由于AVL树，但是当百分比过大时，这一优势不但消失而且时间性能不断增大并且远远高于AVL树，分析可知，这是由于splay树的splay操作过于复杂而在测试机跨度过大时浪费了过多的非查找所用的时间。同时，n与m取的数值是否合适也将影响结果的正确性。但是，从上面的数据也可以发现一些问题。比如，根据理论情况，在数据很有局部性并且比较集中的时候数据应该是splay树明显低于AVL树。但是这一特点在我的数据集中体现的并不明显。这可能是由于，splay树算法虽然是基于AVL树的，但是具体的实现细节可能并不相同，导致了一定的理论与实际之间的误差。

对splay树的理解：

Splay树设计的算法最适合的情况是进行跨度较小的、数据较为连续的查找操作。个人认为splay树的可能的使用情况为：在某些数据结构内部进行连续的查找取值操作。同时，splay树也可能在现代的大数据分析处理（如从大量数据中举出具有相似特征的数据点等）情况中，有较为广泛的应用。