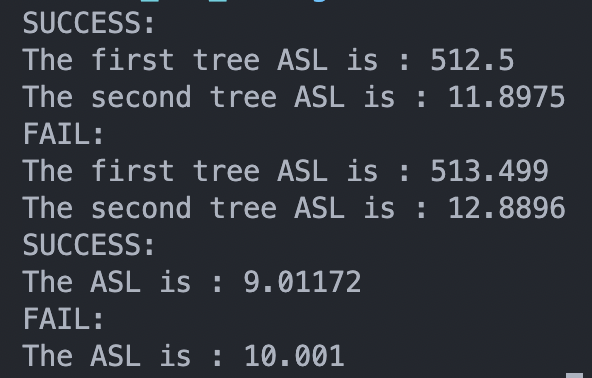
关于平均查找长度：

1. 对于查找成功，模拟了所有可能的查找成功情况，并假定各种情况发生的可能性是相同的。对于查找失败的情况，假定出现在已排序好的节点间间隙的概率相等。
2. 对于用已排序好的数组构建的BST查找树，可以预测，成功的平均查找长度约为

(1024+1)/2 = 512.5。而失败的平均查找长度约为 (1024+1)/2 + 1 = 513.5。（这里由于是已经排好序的，故于在无序的单链表上查找有所区别）

1. 对于未排好序的数组构建的BST查找树，可以假定此查找树相对均衡，成功的平均查找长度大约为 log21024 = 10，而失败的平均查找长度大概为 log21024 + 1 = 11。
2. 对于折半查找，可以发现，折半查找成功的平均查找长度约为log21024 – 1 = 9，而失败的平均查找长度约为log21024 = 10。
3. 测试结果：



这时发现，BST树的平均查找长度并没有接近折半查找的平均查找长度，与BST的最好查找长度略有一段距离。这可能是由于数据并没有完全分散开来，使得BST树并不均衡导致的结果。

1. 上述实验表明：BST树的查找长度还是与折半查找差不多，这是由于BST树将数据组织成一个树的形式，使得查找每一个数据所需要的比较次数，接近 log2n, 这就导致其平均查找长度与折半查找差不多。

但就性能而言，BST树还是与折半查找法有着一点差距，这是由于输入BST树的数据具有随机性，这无法使得BST树组织成为最均衡的状态，导致可能比理论上的最短多一些。（这时可能需要AVL树，进行调整后实现均衡）这能说BST树的查找算法与折半查找很接近，但仍有微小差距。