

实验日志

时间：2019.12.16，19：00-24：00

实验内容：基于顺序查找与基于 BST 的查找的查找算法实现与性能分析

实验过程：

- 1、生成查找数据
- 2、编写顺序查找文件并执行
- 3、遇见问题一
- 4、问题解决再次执行
- 5、遇见问题二
- 6、编写基于 BST 的查找文件并执行
- 7、分析结果

遇到的问题及解决：

问题一：每次查找次数由 1 到 n 递增

由于查找与生成查找文件都是调用的随机函数 rand()，而该函数是伪随机数，多次运行时，每次取值都是一样的，所以查找时随机生成的数和查找文件里的数是一样的。

解决方法：

因此使用 std::e()来生成随机数，改变生成种子就能解决该问题。

错误二：查找时间太小，用 time()测出来都为 0

小数据量查找时，查找时间果断，time()函数的精度无法测出时间。

解决方法：

改用 QueryPerformanceFrequency(&nFreq);函数测出查找执行时,CPU 时钟的跳变周期数，然后除以 CPU 频率，测出执行时间。

结果分析

实验结果

顺序成功查找结果

	最大查找时间(μs)	最小查找时间(μs)	平均查找时间(μs)	最大查找次数(次)	最小查找次数(次)	平均查找次数(次)
100	0.9	0.1	0.395	100	2	47.17
1k	16.6	0.2	4.272	988	2	499.09
10k	83.3	0.8	41.168	9815	74	5047.118
100k	226.6	2.7	90.39804	99506	1394	44664.87

1m	2349.1	3.3	1054.378	943659	1670	498549.2
----	--------	-----	----------	--------	------	----------

顺序查找的失败查找结果

	最大查找 时间(μ s)	最小查找 时间(μ s)	平均查找 时间(μ s)	最大查找 次数(次)	最小查找 次数(次)	平均查找 次数(次)
100	0.8	0.5	0.605	100	100	100
1k	16.1	7.9	8.215	1000	1000	1000
10k	58.4	19.8	20.391	10000	10000	10000
100k	321.6	198.8	202.378	100000	100000	100000
1m	2744.4	1994.9	2021.944	1000000	1000000	1000000

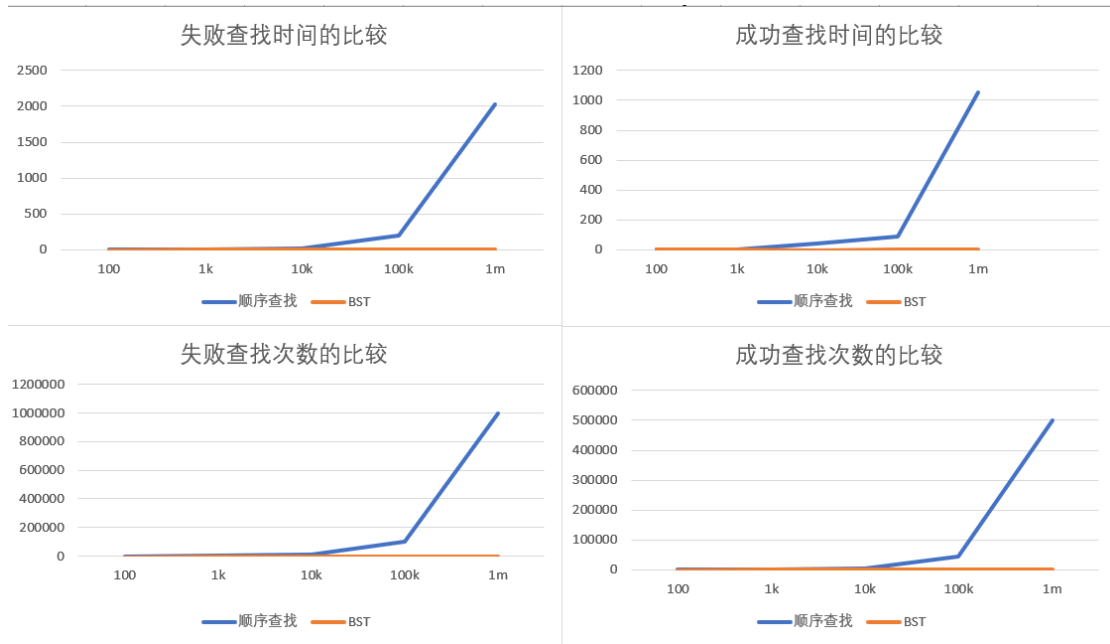
基于 BST 查找的成功查找结果

	最大查找 时间(μ s)	最小查找 时间(μ s)	平均查找 时间(μ s)	最大查找 次数(次)	最小查找 次数(次)	平均查找 次数(次)
100	0.5	0.1	0.33	7	1	4.43
1k	0.9	0.2	0.577	10	1	5.71
10k	0.54	1.4	0.1	14	1	7.16
100k	1.8	0.2	0.74	2	17	8.27
1m	1.8	0.2	0.746	20	1	9.78

基于 BST 查找的失败查找结果

	最大查找 时间(μ s)	最小查找 时间(μ s)	平均查找 时间(μ s)	最大查找 次数(次)	最小查找 次数(次)	平均查找 次数(次)
100	5.3	0.4	0.526263	7	6	6.666667
1k	1.2	0.8	0.932	10	9	9.99
10k	1.1	0.7	0.878	14	13	13.46
100k	5.4	0.9	1.221	17	16	16.85
1m	3.7	1	1.255	20	19	19.82

结果分析



实验结论

由以上的数据显然可见，再小数据量的情况下，BST 与顺序查找差距不大，但在大数据量的情况下，BST 可以比顺序查找快相当多。

BST 查找的时间复杂度为 $O(\log n)$ ，而顺序查找时间复杂度为 $O(n)$ ，根据理论计算，在一百万数据量的情况下，顺序查找的查找次数应为 BST 的 5 万倍，该实验的测出为 5 万倍，与理论相符很好。