RITA C. C. CRUZ
20200041181
RCCHCZ@GMAIL.COM

1 Is it really true that the binary search is more efficient than the linear search?

Empirically, the binary search was more efficient in all tests.

Tests were performed from 100000 to 950000 elements, adding 50000 at a time. It is worth noting that each time spent in each test of n elements was actually originated from the average of 10 tests for these n elements. Furthermore, the tests were performed on a vector of 1000000 elements, all 0. Since efficiency was measured based on the worst case, the target-value was 1.

When testing the iterative versions of the linear and binary search algorithms, the first test case referred to 100000 elements. In this case, the linear search was slower (0.19656ms), while the binary search only took (0.00037ms) – a difference of 0.19619ms. In other words, the binary search could be performed approximately 531 times for each linear search for that amount of elements (100000).

For the largest amount of elements (950000), the linear search took 3.52994ms, while the linear search only took 0.00202ms. Analyzing the efficiency difference, the linear search took 3.52792ms. This means that, under these test conditions, the binary search could be performed approximately 1747 times for each linear search.

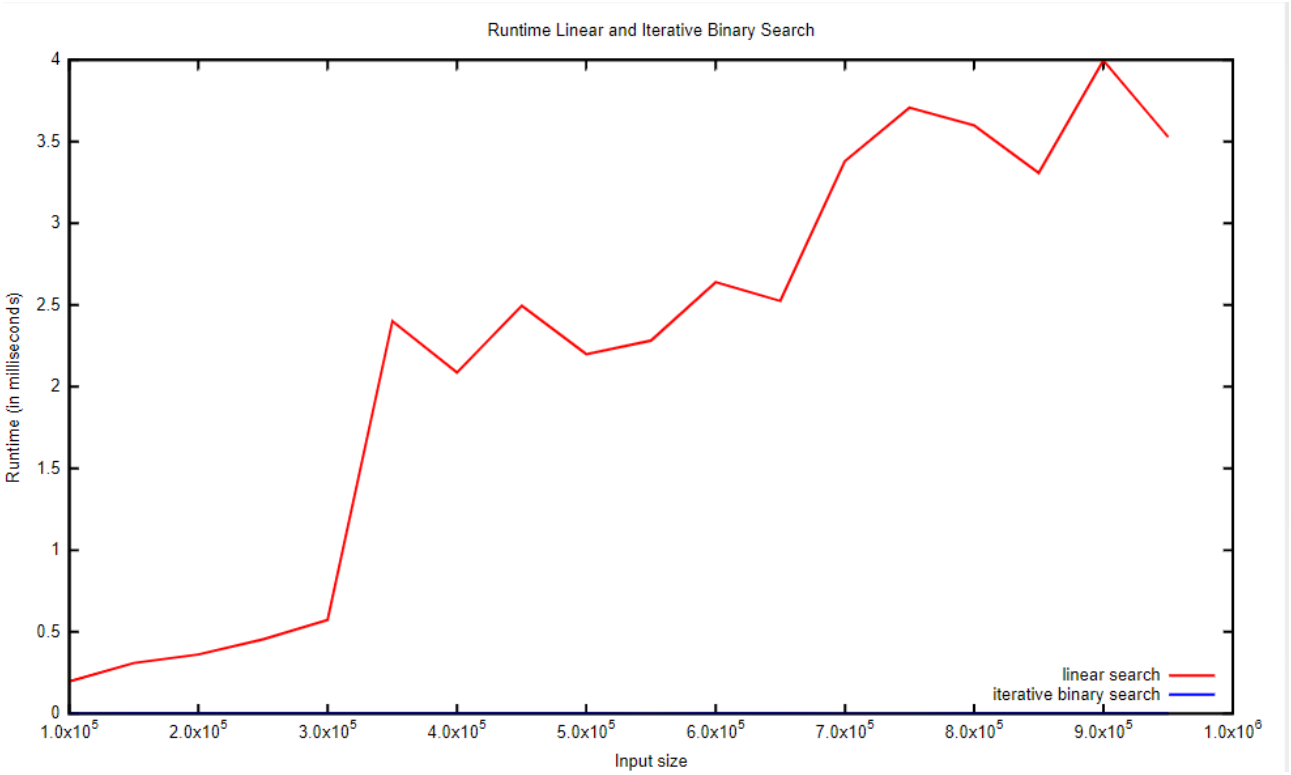
Therefore, looking at the test results, binary search is most efficient in the worst case – which corresponds to searching for a target that is not in the collection.

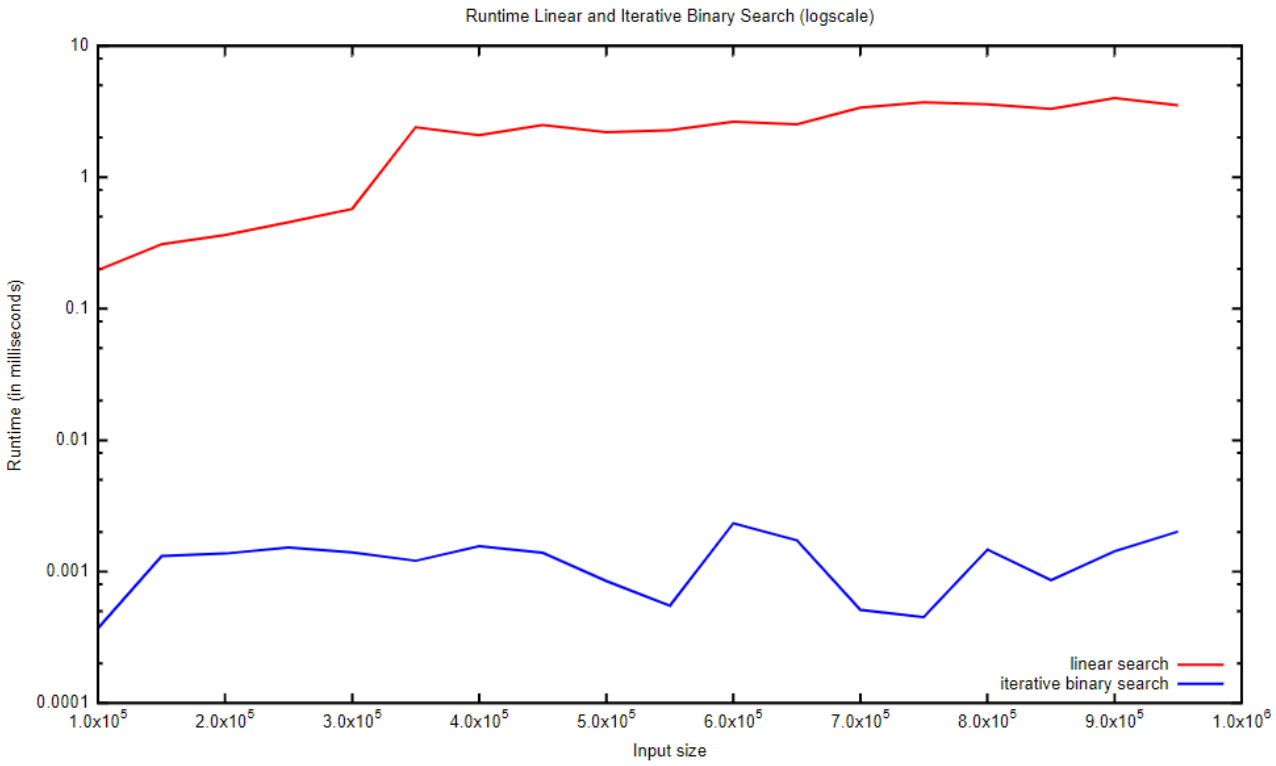
1.1 Linear and binary: sampling table

Input size	Linear search (ms)	Iterative binary search (ms)
100000	0.19656	0.00037
150000	0.30962	0.00131
200000	0.36259	0.00137
250000	0.45518	0.00152
300000	0.57320	0.00140
350000	2.40016	0.00121
400000	2.08691	0.00156
450000	2.49525	0.00139
500000	2.19856	0.00085
550000	2.28211	0.00055
600000	2.64047	0.00233
650000	2.52481	0.00173
700000	3.37958	0.00051
750000	3.70648	0.00045
800000	3.59781	0.00147
850000	3.30788	0.00086
900000	3.99517	0.00143
950000	3.52994	0.00202

1.2 Linear and binary: runtime graphics

Because of the graphics, the complexity of the algorithms can be better understood: O(n) for linear search, O(log n) for binary search. The efficiency of each search algorithm can be verified in the graphs below.





2 Which one is more efficient, the iterative or the recursive version of the binary search?

Despite similar runtimes, the recursive version was more efficient. The same method commented above was applied to the efficiency tests for the iterative and recursive binary search algorithms.

In the first test (100000 elements), the iterative search was more efficient – a fact that was repeated in other tests. However, the possible interference of other running routines, that is, browser and other applications, must be recognized. This is because, in most tests, the recursive binary search proved to be more efficient,

taking less execution time.

2.1 Linear and binary: sampling table

Input size	Iterative binary search (ms)	Recursive binary search (ms)
100000	0.00037	0.00043
150000	0.00131	0.00053
200000	0.00137	0.00047
250000	0.00152	0.00051
300000	0.00140	0.00062
350000	0.00121	0.00053
400000	0.00156	0.00055
450000	0.00139	0.00061
500000	0.00085	0.00071
550000	0.00055	0.00061
600000	0.00233	0.00093
650000	0.00173	0.00066
700000	0.00051	0.00067
750000	0.00045	0.00083
800000	0.00147	0.00093
850000	0.00086	0.00064
900000	0.00143	0.00065
950000	0.00202	0.00066

2.2 Iterative binary and recursive binary: runtime graphics

The graph proves the equal time complexity of both methods of implementing the binary search: O(log n). However, it is worth noting the slight advantage of the recursive implementation.