

Problem: 3.1.40 - Crossover to binary search
[Taken from the book Algorithms by Sedgewick, Wayne]

Roll No: 20186097

Name : Rohith

Analysis 3.1.40 - Crossover to binary search

Analysis

Considering that sequential search has a complexity of $O(n)$ and binary search has a complexity of $O(\log(n))$:

N = 58: While sequential search does 58 operations (in the worst case), binary search does ~5.8 operations (in the worst case).

Binary search is ~10 times faster than sequential search for N = 58.

N = 996: While sequential search does 996 operations (in the worst case), binary search does ~9.96 operations (in the worst case).

Binary search is ~100 times faster than sequential search for N = 996.

N = 13746: While sequential search does 13,746 operations (in the worst case), binary search does ~13.746 operations (in the worst case).

Binary search is ~1000 times faster than sequential search for N = 13,746.

Experimental results:

(Java.util.Random class was used to put and get random values in a random order.. May vary each time, you run the program.)

- Value of N for which binary search becomes **10 times** faster than sequential search: **120**
- Value of N for which binary search becomes **100 times** faster than sequential search: **1463**
- Value of N for which binary search becomes **1000 times** faster than sequential search: **22461**

Average and Worst case scienario

	10 x faster	100 x faster	1000 x faster
Average case	Sequential Search = [2, 61] Binary Search = [1, 9]	Sequential Search = [3, 447] Binary Search = [4, 17]	Sequential Search = [198, 1809] Binary Search = [7, 20]
Worst case	Sequential Search = 62 steps. Bianry Search = 9 steps.	Sequential Search = 841 steps Binary Search = 19 steps.	Sequential Search = 13402 steps. Binary Search = 26 steps.

Eg: Sequential Search [2, 61] means that for a randomly generated symbol table when getValue() function is randomly called it took a minimum of 2 steps and a maximum of 61 steps over 50 iterations.

Inference

Binary search is efficient only when the size of the symbol table is very big. (Probably in terms of 100's). Because binary search symbol table needs keys to be in sorted order. And if the sorted order is preferred, then sequential search symbol table outperforms binary search symbol table (Since it takes constant time as we can retrieve the value by passing the index of the sorted key array). So, from the random experiment conducted, it can be stated that BST outperforms Sequential search symbol table (Limitation: size of the symbol table should be taken into account).

Java files

Please, look into the readme file to understand the hierarchy of the java files.