Mateusz Małecki - 1482651
Nikodem Seidel - 148269

# Comparison of ordered linked list, binary search tree and balanced binary search tree.

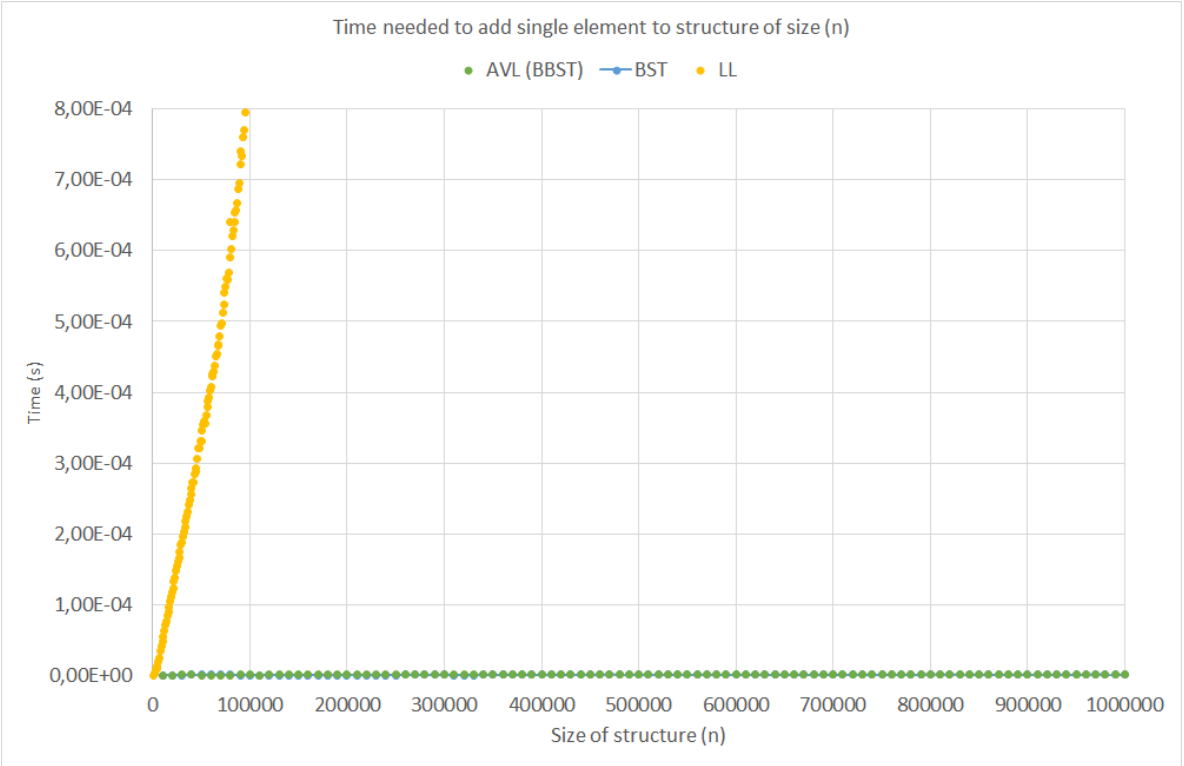
## Ordered linked list

Linked list (LL) is a data structure that includes a sequence of connected nodes. Every node consists of two parts. Stored data and a pointer to the address of the next node. This approach causes that to get value from the 3rd node, one must go through the first and second node thus time complexity of accessing nodes is equal to O(n) (linear). There is no possibility to get value faster (e. g. by random choose). This structure also limits traversing backward. Every element stores only the address of the next node therefore, the address of the previous node is unknown and unreachable.

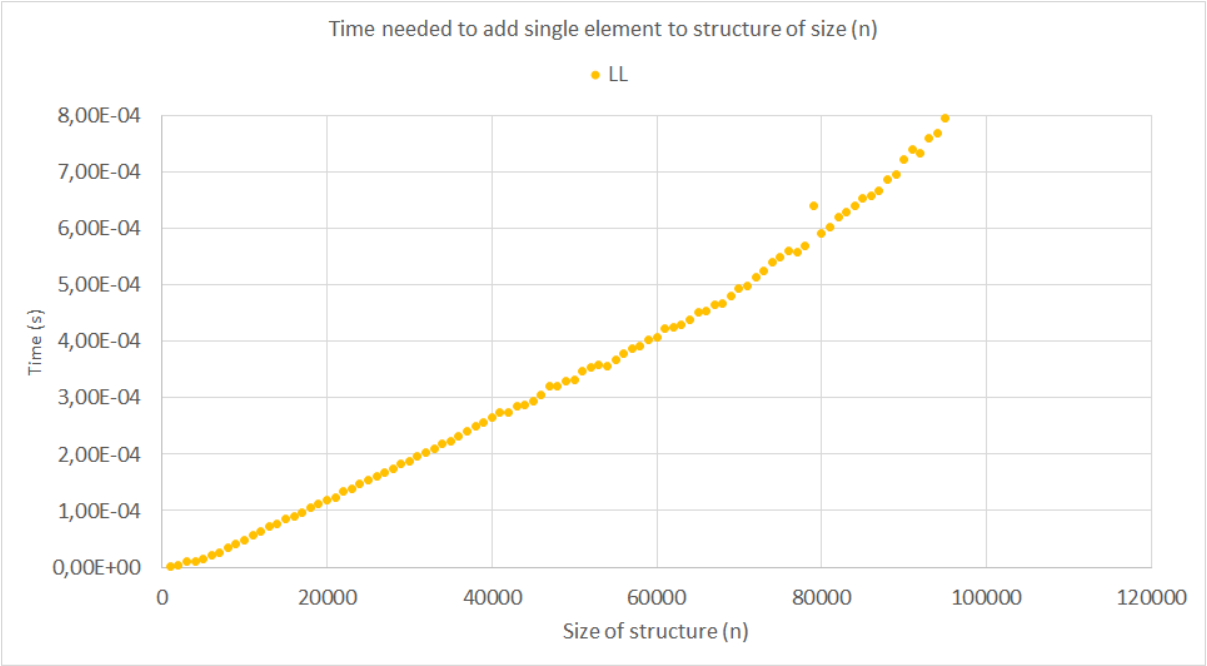
But the Ordered Linked List has a lot of pros. The most important is that it is always sorted. Specific inserting function allows you to keep order without sorting the list after every insert. In contrast to normal arrays, LL allows fast insertion and removal of values in particular places without realocating and reorganization of the entire database. Second one is that it is very easy to understand and implement in a code. Also there is no need to know how big finally this list will be. These advantages make it one of the most common data structures used in the IT world.

Before presenting all graphs we want to explain some important issues.
Remark 1: Adding means insertion of elements to the ordered database in such a way that structure will stay ordered. It's very important because unordered addiction of the element on the first place in the linked list has complexity equal to 1 (constant).
Remark 2: In analysis below all added to the structures data was unordered before use. Searching through and removing elements from the data structures was also totally random.
Remark 3: In case of tree data structures, we've used databases of size 1 000 000, with unique indices. All results of our test are a mean value of 10 runs with different databases. Measurements were taken every 10 000 elements.
Remark 4: Databases for testing Linked List were size 100 000 because testing bigger instances would make no difference in our graphs and would take much time to compute. In this case measurements were made every 1 000 elements. Results are also mean of 10 runs with different databases.

On the chart below the ordered linked list was compared to binary search trees (balanced and unbalanced).



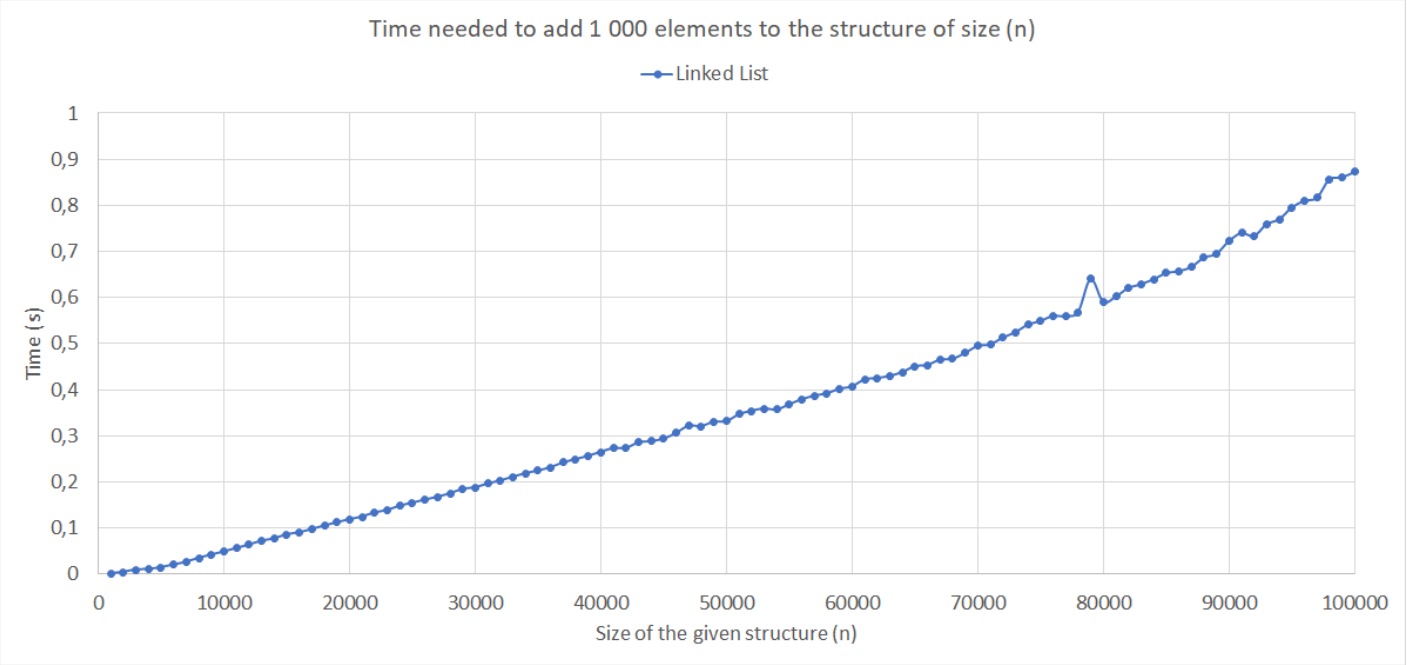

And the same graph with Ordered linked list separately.



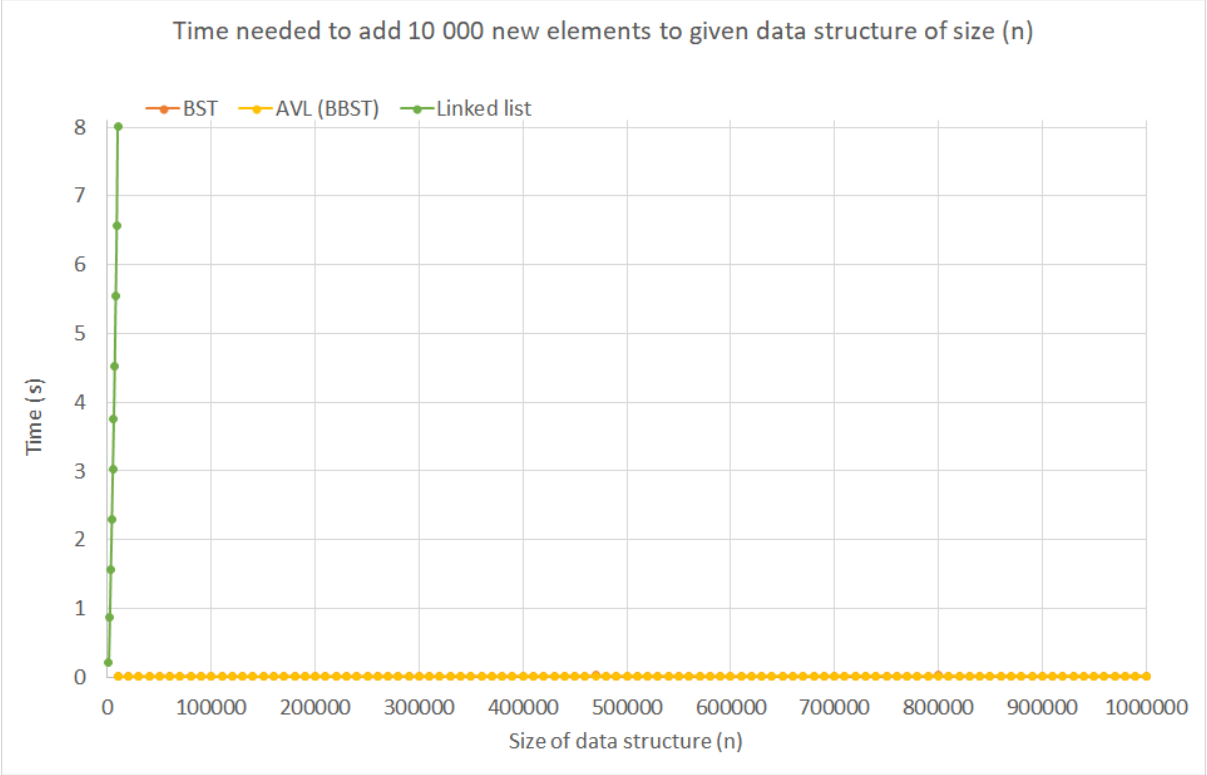

On the first graph, the sorted linked list is significantly less efficient than BST and AVL. That's because in contrast to trees there is only one path that every node follows until the algorithm finds a proper place for it. In case of trees, data is stored in separate branches,

which simply shorten the path to the searched place. Line of the linked list on the chart above is straight (a linear function) so we can conclude the complexity of sorted addiction of elements equals to O(n) where n is the size of the database - which was expected, since every new node in the worst case has to be compared with all nodes of given structure. Worth mentioning is the fact that addiction of the sorted element to the ordered linked list consists of two actions. Searching the right place to insert node (linear complexity) and adding i.e. changing values of the pointers in neighboring nodes (constant complexity). So summarizing, the complexity of this function is equal to complexity of search time plus constant (O(n) + O(1)).

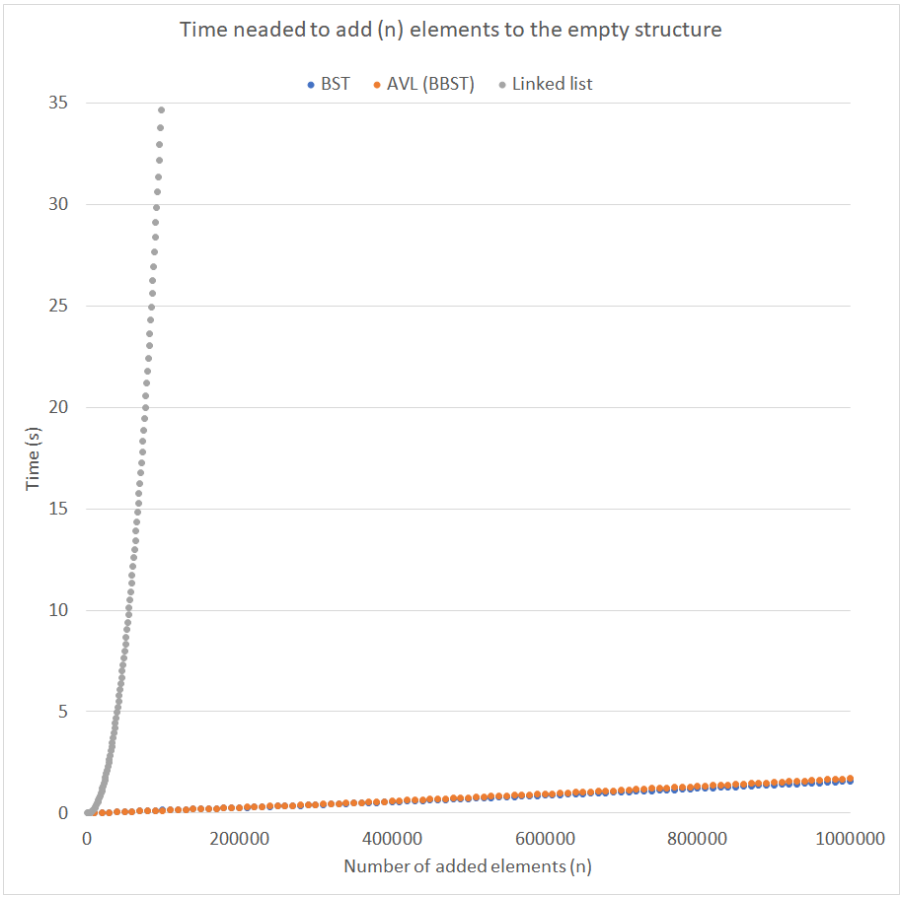
Next two graphs show how much time is needed to add 1 000 (first graph) and 10 000 (second graph) new elements to the particular structures. First consist of only an ordered list.



The second one is a comparison of the list and the trees.

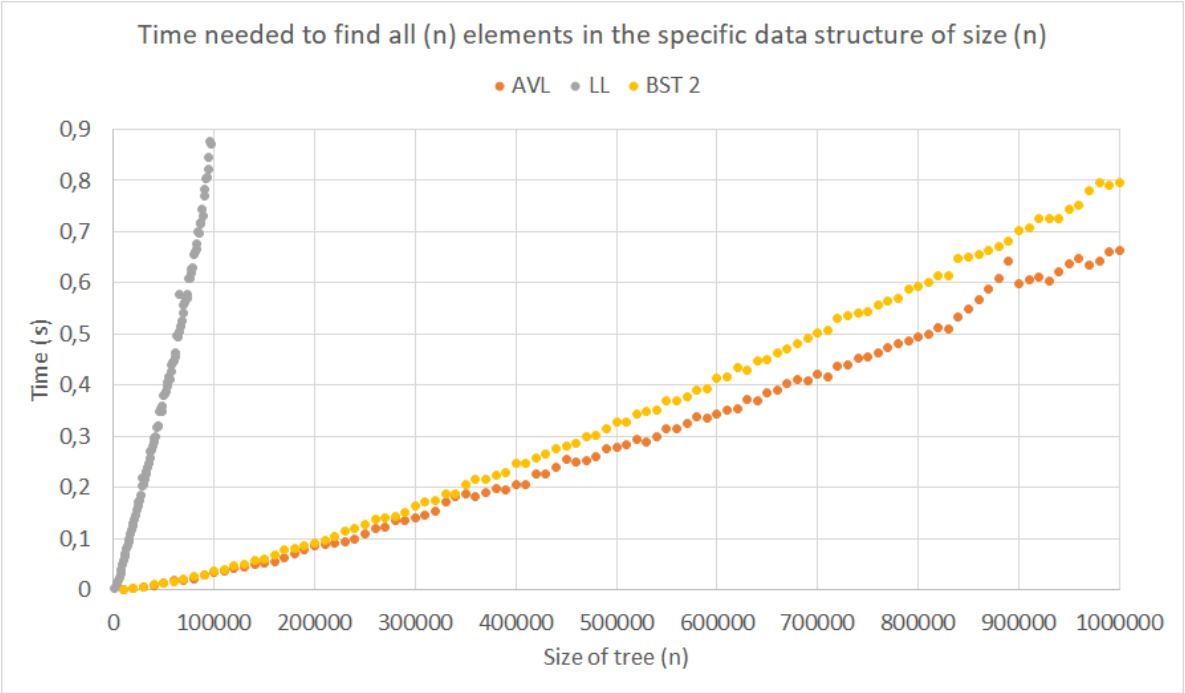


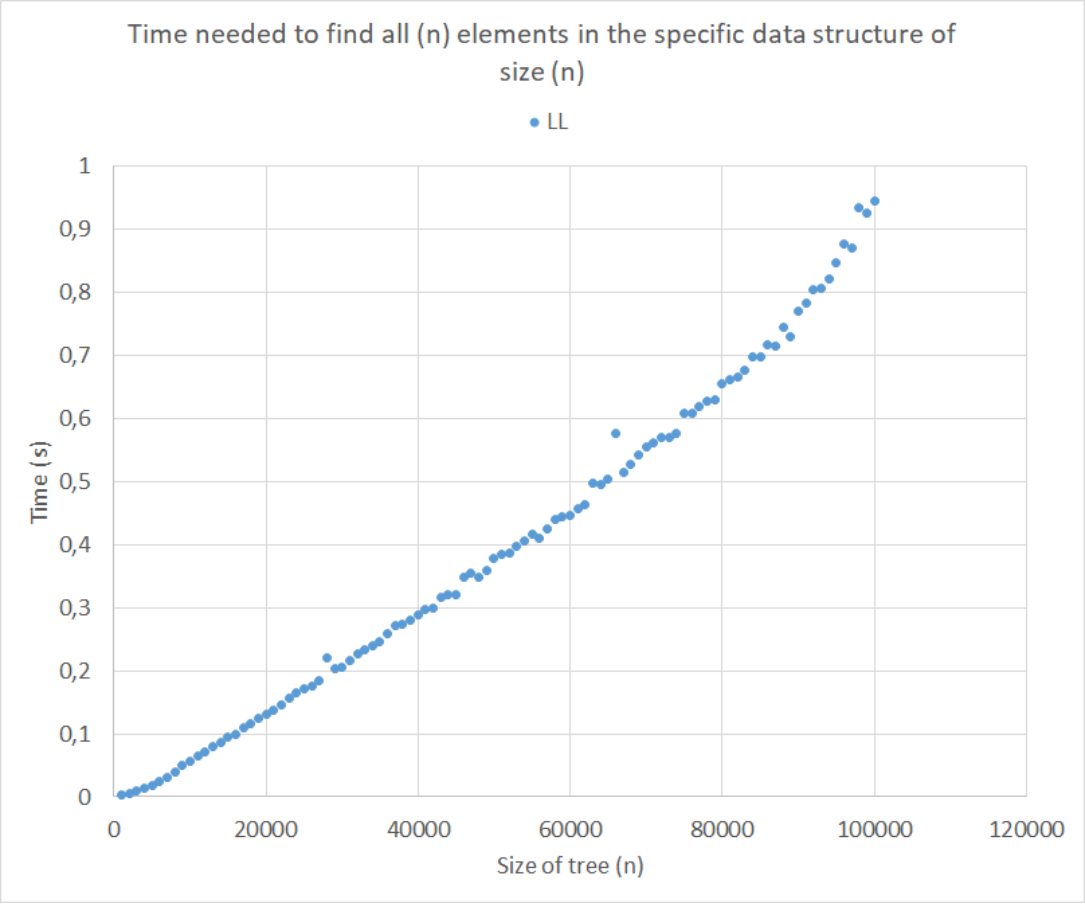

The last graph related to addiction shows time needed to add n elements to the empty structure. To compute complexity of this action we have to multiply complexity of addiction single element by number of elements, thus binary search tree has O (nlog n) and ordered linked list leads to $n^2$.

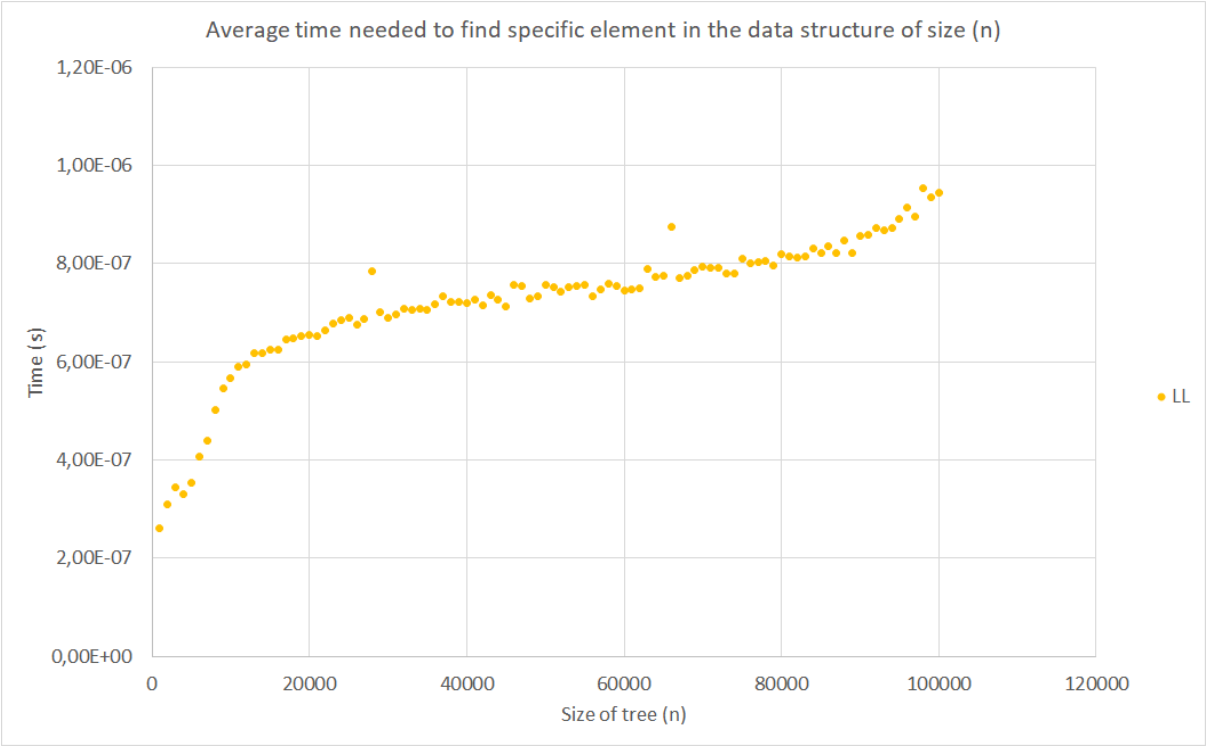


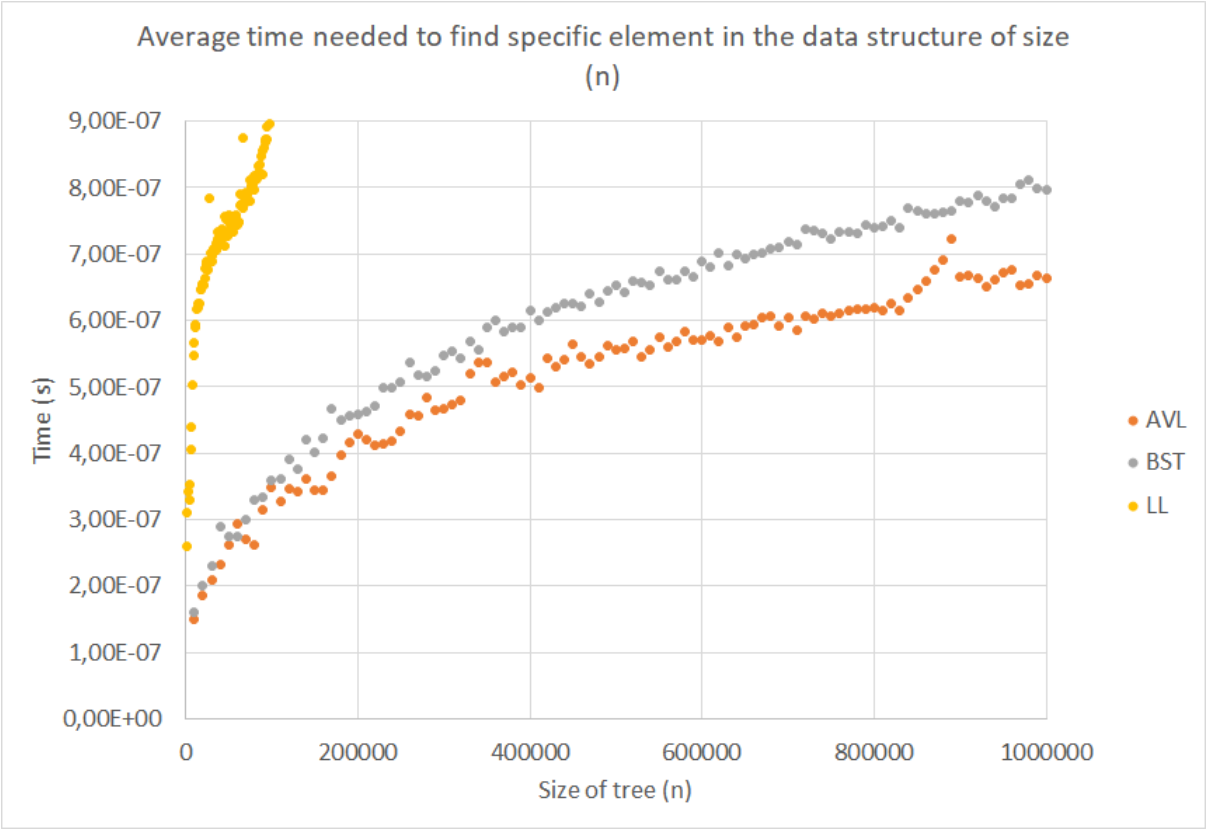
**Searching**

Graphs which represent time needed to search one element in an ordered linked list are pretty similar to charts with addiction and that's because to insert the node in a proper way we have to first find the right place so basically addiction function involves searching. It causes that complexity of this is the same as the inserting function and it's equal to O (n). In comparison to BST linked list is highly inefficient.



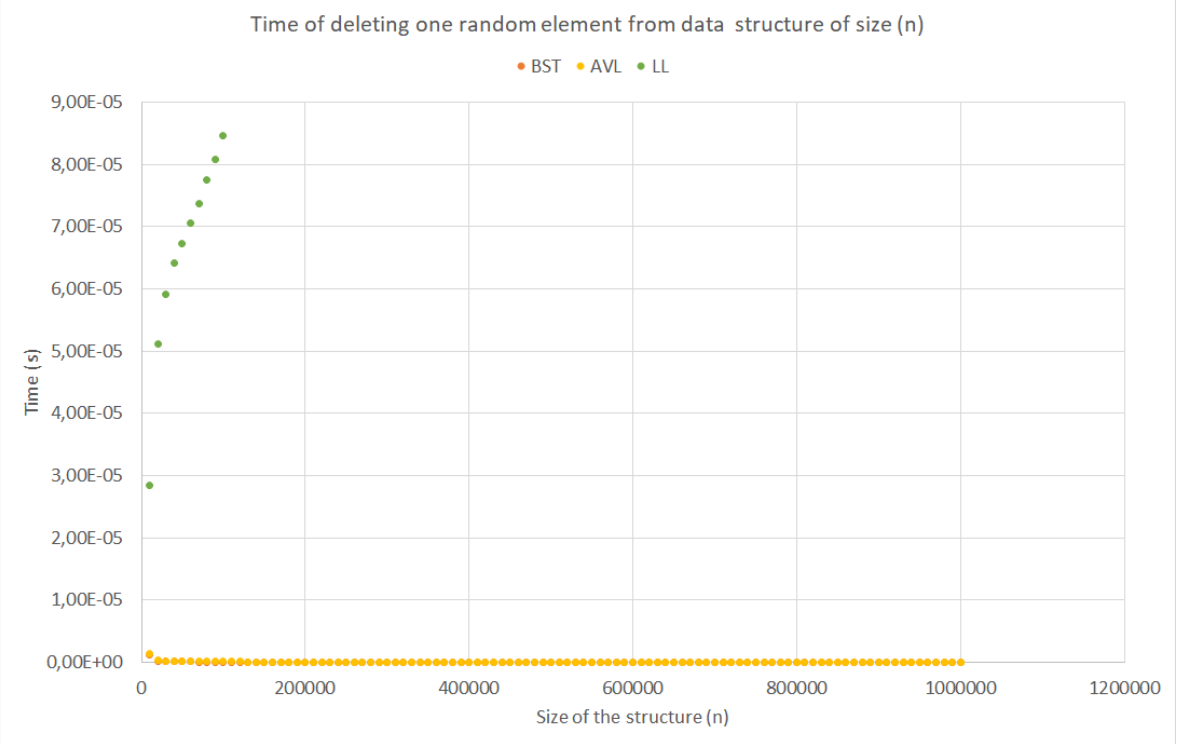

Average time needed to find specific element in the data structure of size (n)
1,20E-06
1,00E-06
8,00E-07
6,00E-07
4,00E-07
2,00E-07
0,00E+00
Time (s)
0
20000
40000
60000
80000
100000
120000
Size of tree (n)
LL

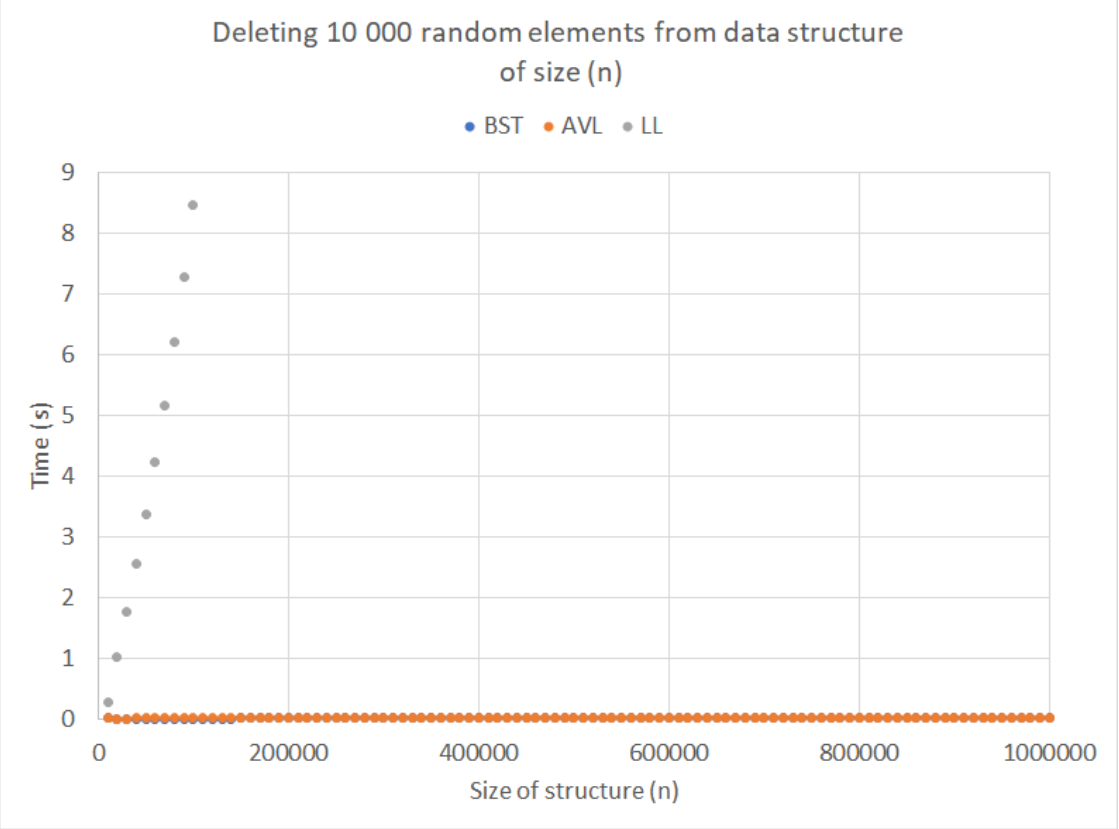

Average time needed to find specific element in the data structure of size (n)
9,00E-07
8,00E-07
7,00E-07
6,00E-07
5,00E-07
4,00E-07
3,00E-07
2,00E-07
1,00E-07
0,00E+00
Time (s)
0
200000
400000
600000
800000
1000000
Size of tree (n)
AVL
BST
LL

**Deletion**

As well as addiction, the removal algorithm uses the searching part first. So it is expected that it has similar time needed to perform and what follows from that it has similar complexity O (n). Also as in previous cases deleting is drastically slower than BST.

Situation changes if the address of the node that has to be removed is known, then complexity isn't dependent on size of database and is equal to 1 (constant).



**Binary Search Tree (BST) and Balanced Binary Search Tree (Adelson-Velsky and Landis tree - AVL):**

BST and BBST is a data structure which was modelled to be fast. It takes the advantage from the easy rule: smaller values go to the left branch, and rest to the right branch, and this condition is considered in every node. In that way, while considering the perfect tree the length of the path to every single node is less than, or equal to $\log_2(n)$, where n is the size of the structure. For the sake of illustration, in the case of a database of size 1 000 000, length of the path to the last node in the linked list is equal to 1 000 000, and in a well distributed binary tree it is non longer than 20 steps. Seems that this structure has no downsides, but there is a dark side of this way of keeping data.
In the case of standard BST, performance of this structure is strictly related to the order in which the data was added / inserted, because if data is sorted, then suddenly BST becomes Ordered Linked List which diametrically decreases speed of adding / searching / deleting nodes.
When it comes to code implementation, a tree is constructed of nodes (structures) which contains some data, and two pointers to the next two childrens (smaller valued - left and bigger valued - right nodes). In the standard version of BST there is no way to iterate the other way than from the root node. Basic operations implemented to BST:

Adding - this operation is pretty easy. Just simply follow the path checking if the value to be added is bigger or smaller than nodes on the path. Complexity of such an operation is equal to $O(\log_2(n))$ where n is the size of the structure.

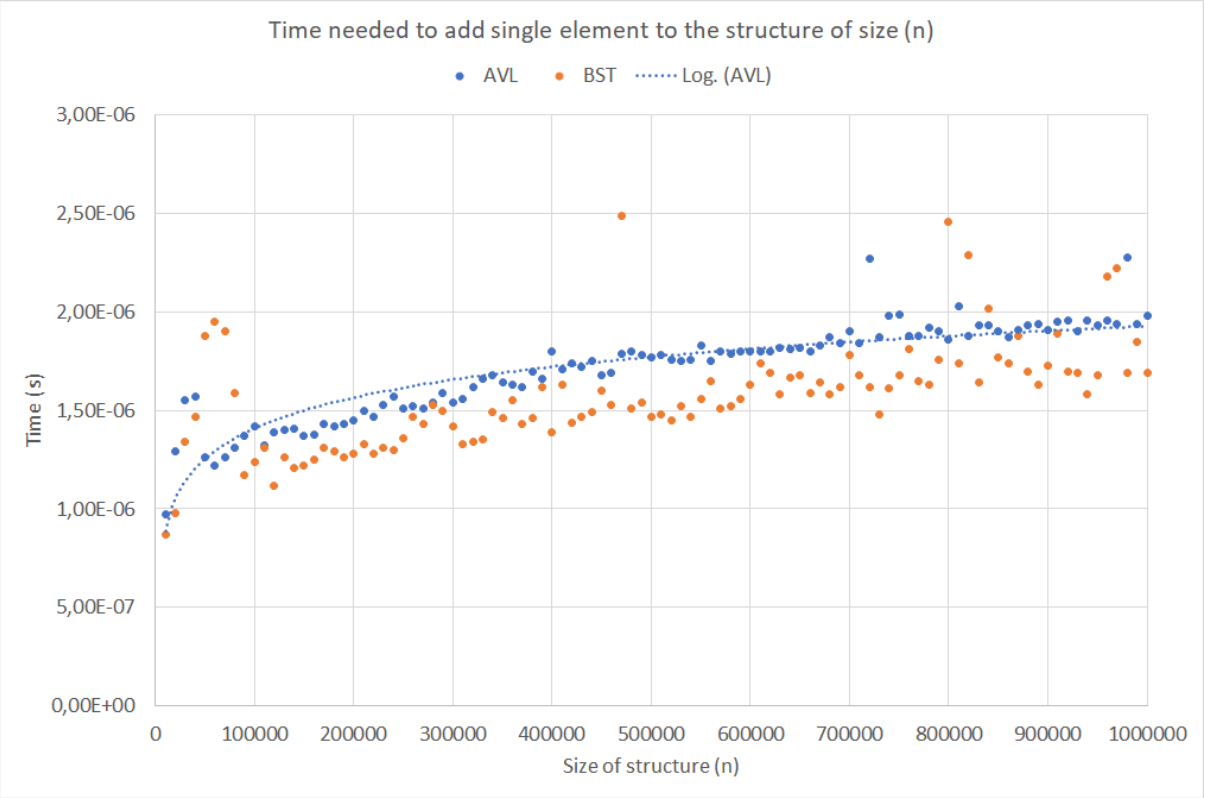
Deletion - this kind of data structure allows us to remove any node we want, but implementation is not that easy. Programs have to know if the node to be deleted has children, and how many. In case of no children or one child, it can be simply deleted and the previous node will have an updated pointer to the considered child. When there are 2 children, the node to be deleted needs to be replaced by a node which is bigger, but as close as possible to the value of the node to be deleted. Complexity of such an operation in the worst case is equal to O(n) (BST becomes LL), and the average complexity is equal to $O(\log_2(n))$ where n is the size of the structure.

Searching is usually a recursive function, which searches from the beginning of the tree to the expected node. Complexity of searching in the worst case is equal to O(n) (BST becomes LL), and the average complexity is equal to $O(\log_2(n))$ where n is the size of the structure.

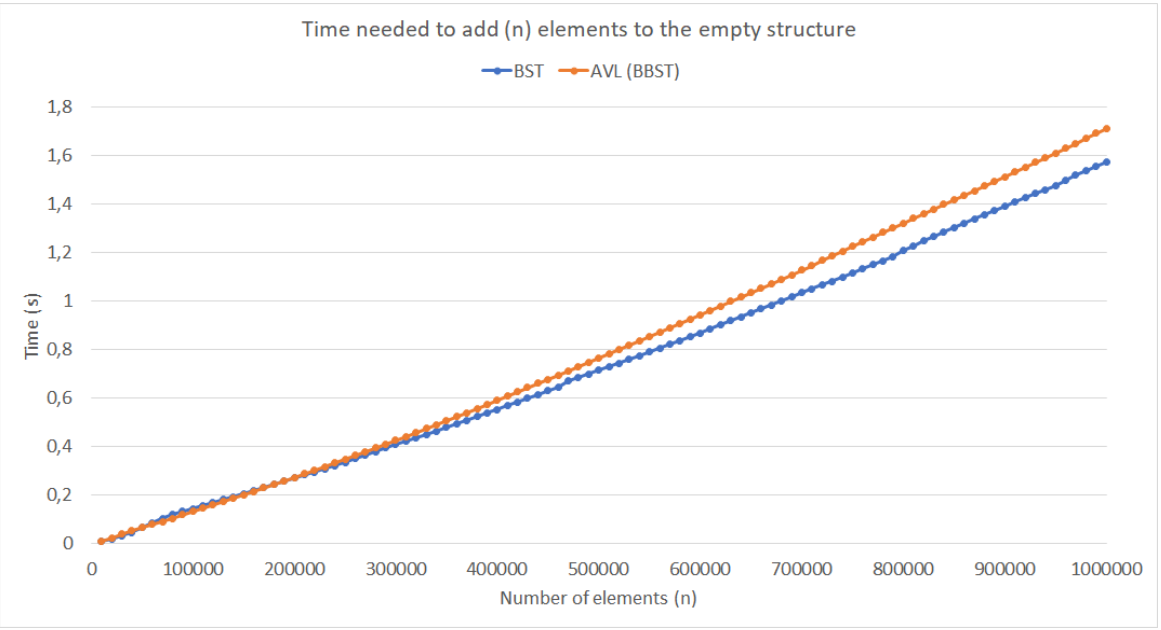
**BBST** is almost the same structure as the BST. The only difference is the fact that after every insertion or deletion algorithm checks if the balance of the tree is kept (in our implementation the difference between height of all leafs had to be non greater than 1. If the balance is disrupted, then the tree performs some rotations of branches to finally become balanced again. This operation takes some time, what can be noticed on the graph below.

*Remark: On the plots above, it can be clearly seen that trees are incomparably faster than the linked list, therefore following graphs will contain only trees.*

**Adding / inserting**



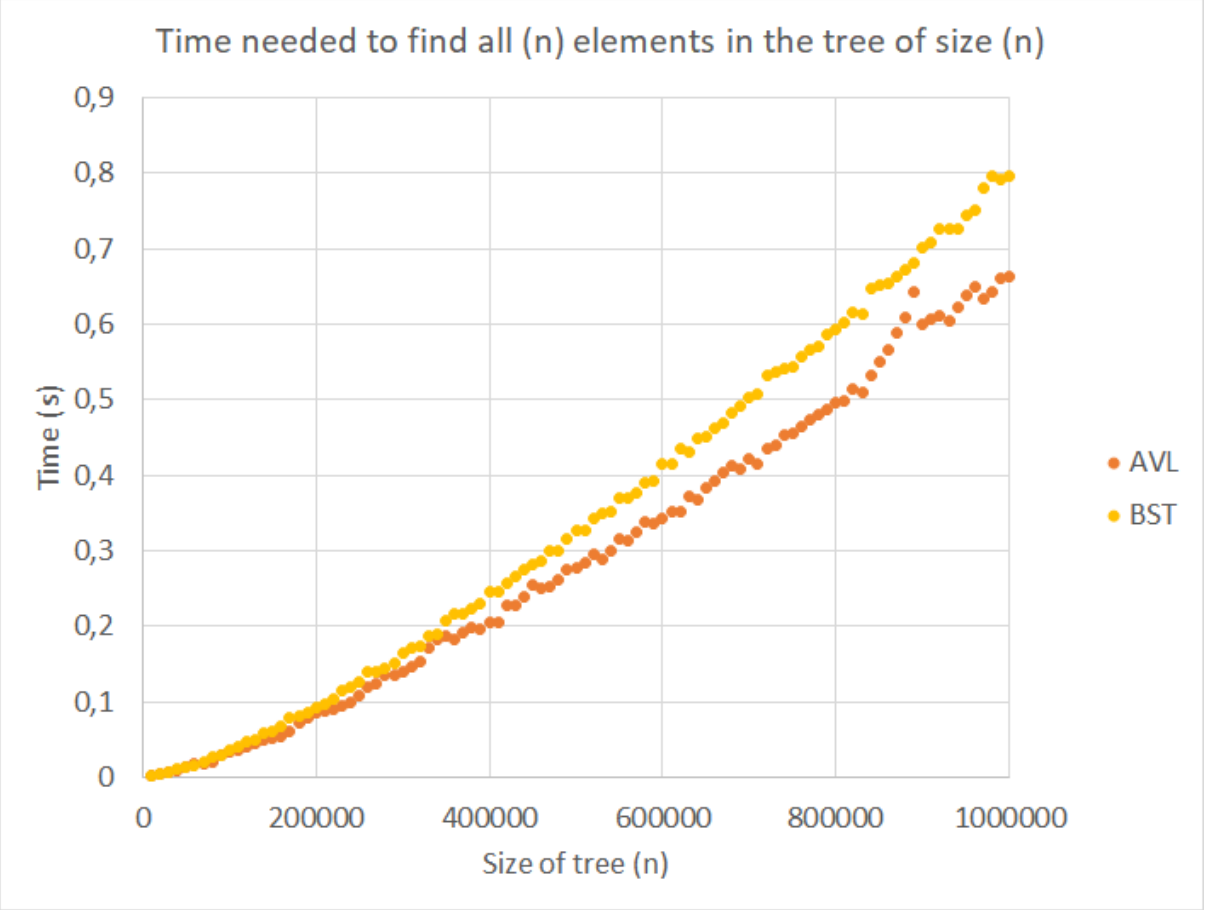

On the plot above, we can see that checking balance after every insertion, and balancing the tree makes AVL slower in terms of adding elements to the structure. As expected since adding element to tree structure has complexity equal to $O(\log_2(n))$, both graphs are lying about logarithmic function.



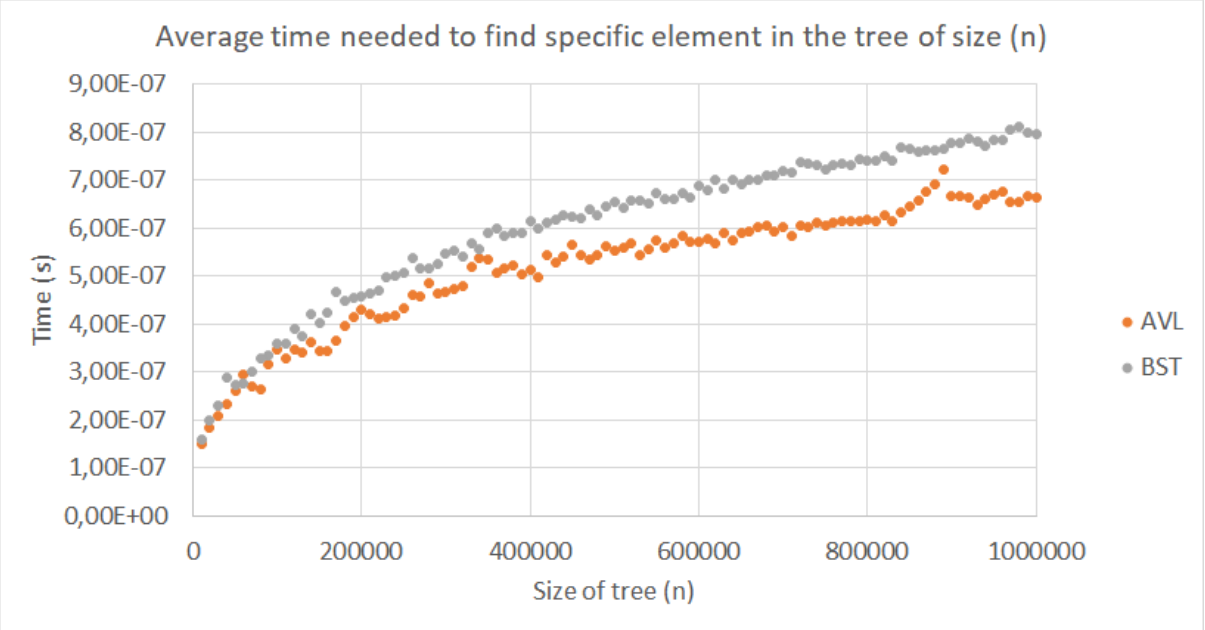
Same property (influence of balancing the tree) can be seen on the graph above. The bigger the structure is, the bigger is the time difference.
As expected, in the average case scenario standard BST is faster in terms of insertion of the new elements.

## Searching:

Let us compare searching of the elements:



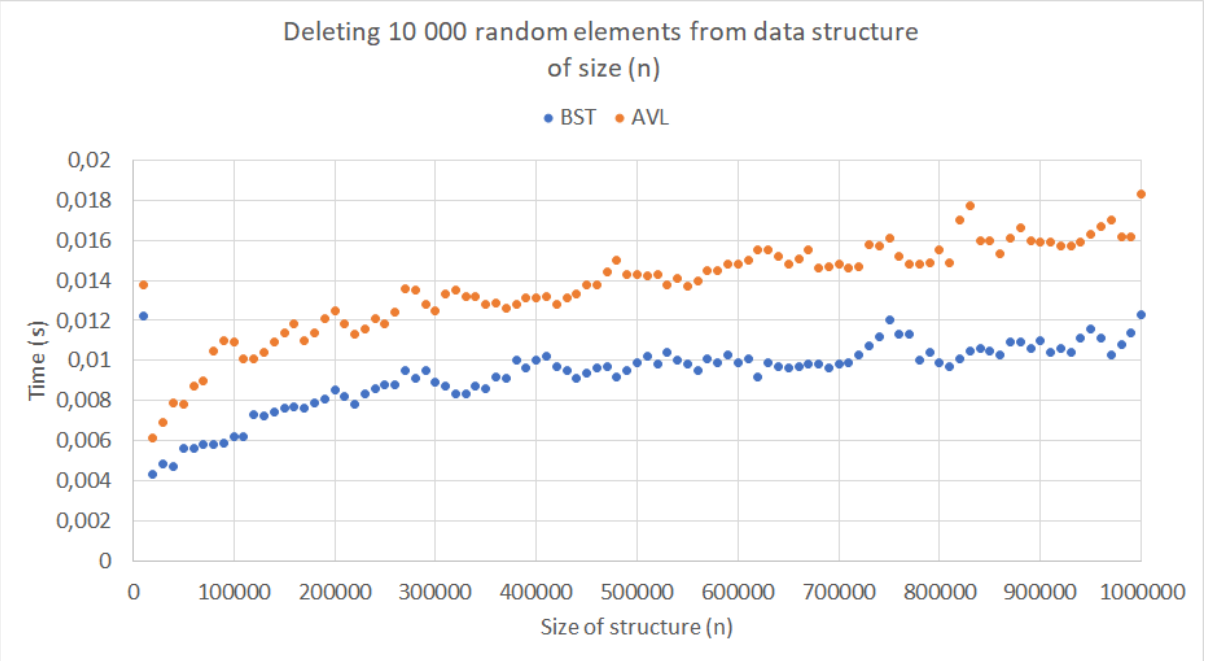

Now the balanced search tree takes an edge of advantage. Equal length of paths to the leaves makes AVL easier to search through, when in the BST lengths of branches are unknown and very likely unequal. Such an operation (searching all (n) elements in structure of size (n)) has a time complexity $O(n*\log_2(n))$ which holds on our graph, since the shape of those functions are same as $y = n * \log_2(n)$
*Graph of the mentioned function* --> (https://www.desmos.com/calculator/soejlimzkv).




Here, we can see that finding the specific element in structure has time complexity equal to $O(\log_2(n))$ as mentioned in the introduction. Also, a balanced tree is faster to search through.

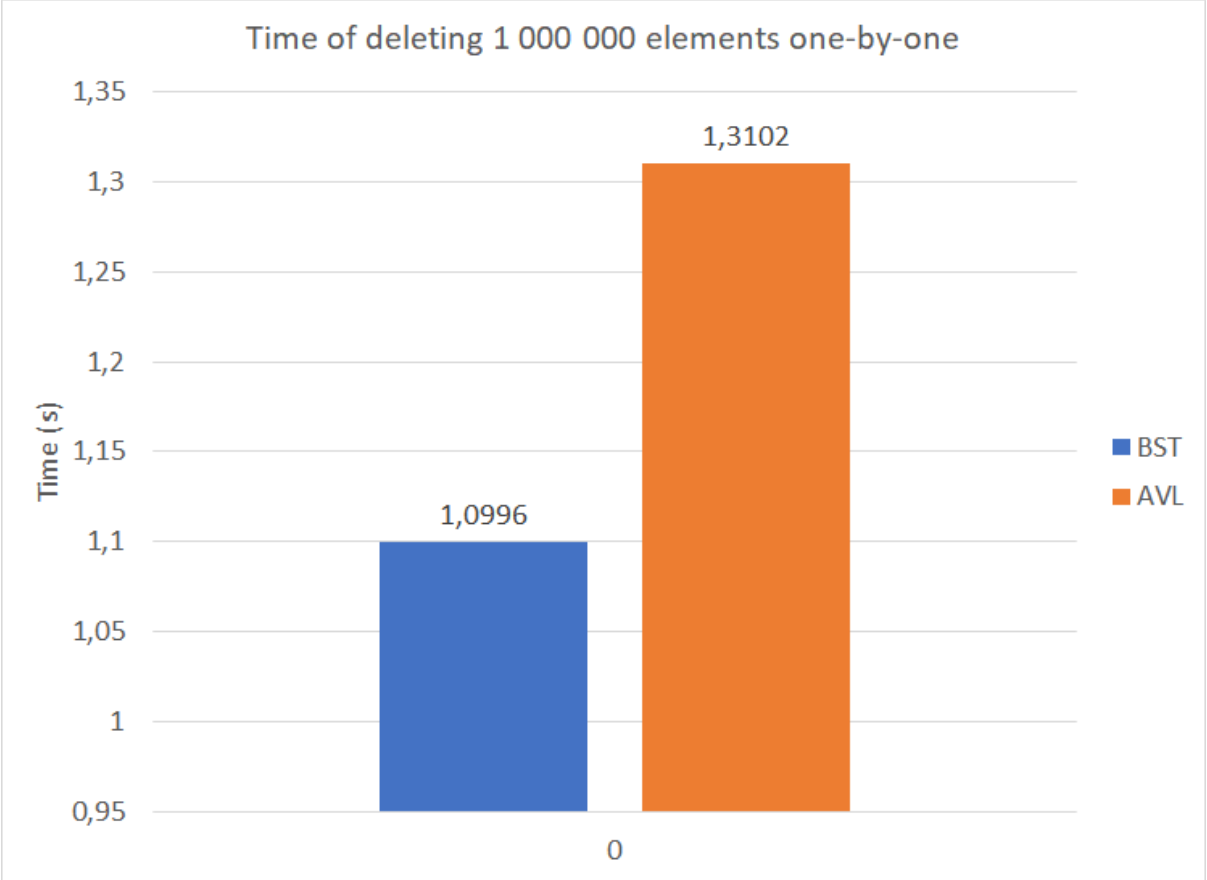
## Deletion:

Deletion in AVL is more complex than in BST because of the need to balance the tree after operation, which can be clearly seen on the graph below.




Deletion of “k” elements from the structure with “n” elements has time complexity of $O(k*\log_2(n))$.

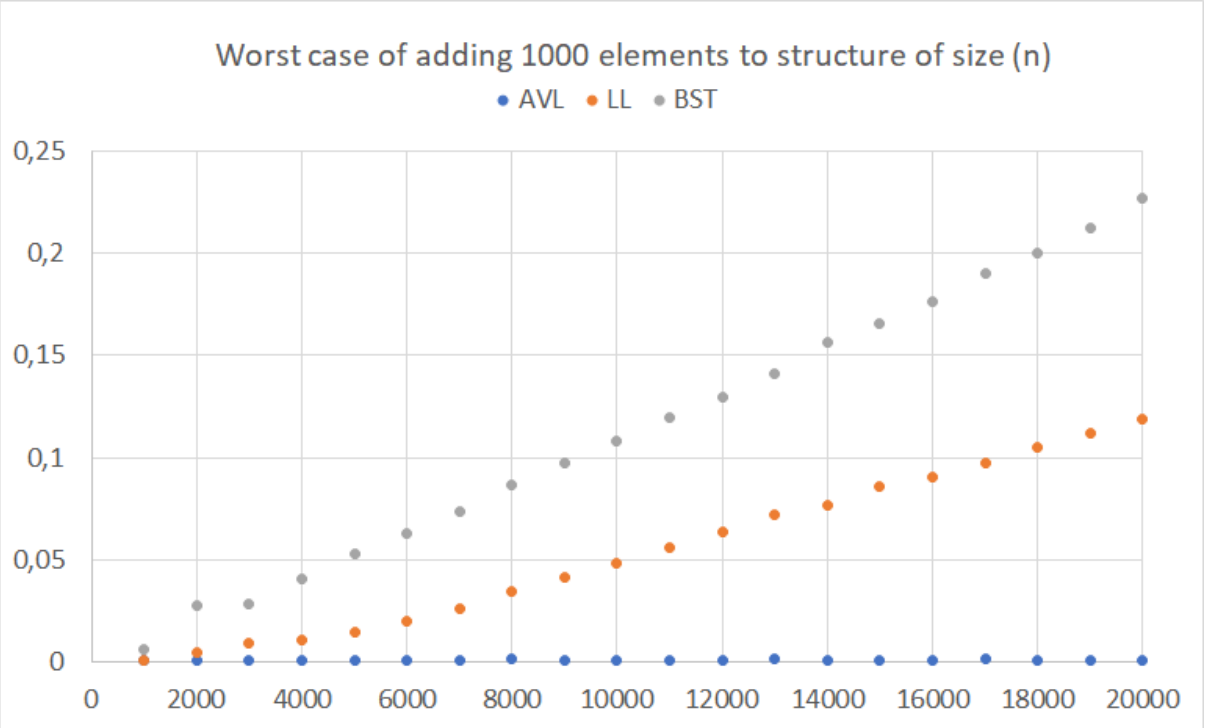
In our test, with 1 000 000 randomly inserted elements, deletion of all elements one by one in random order leasted:

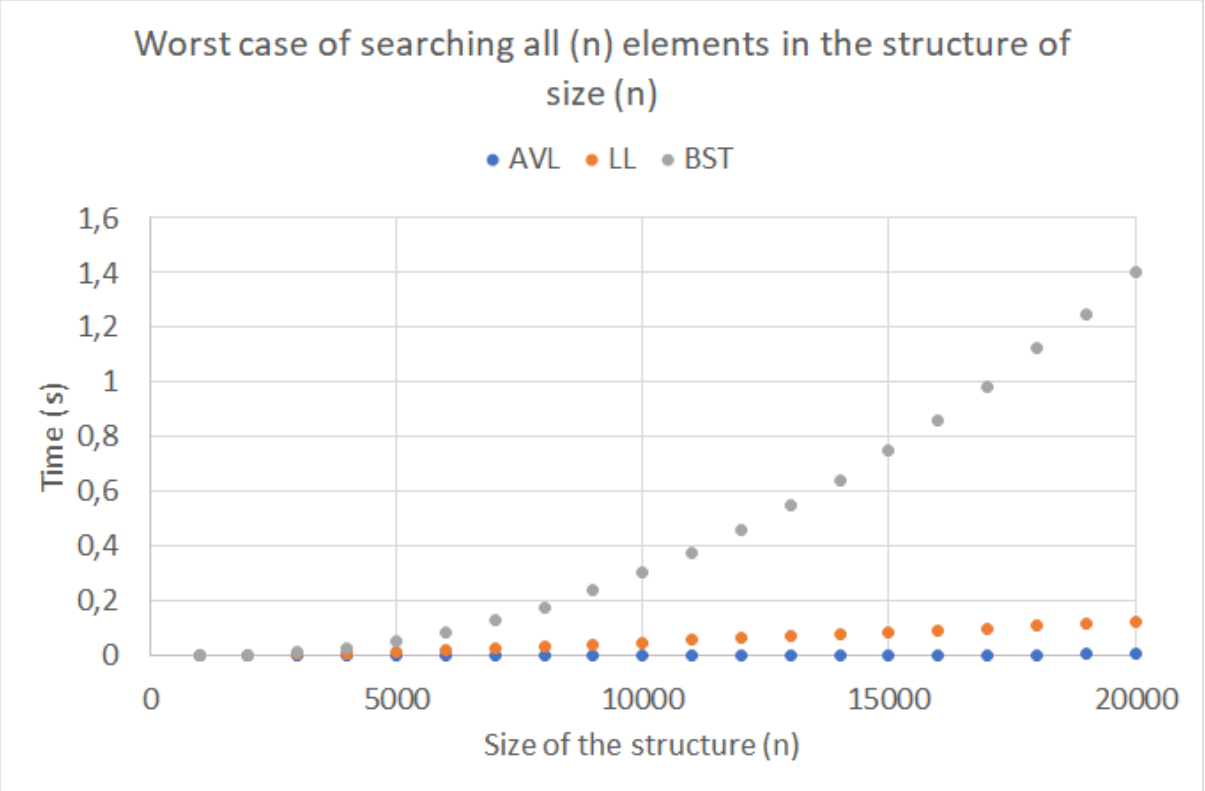


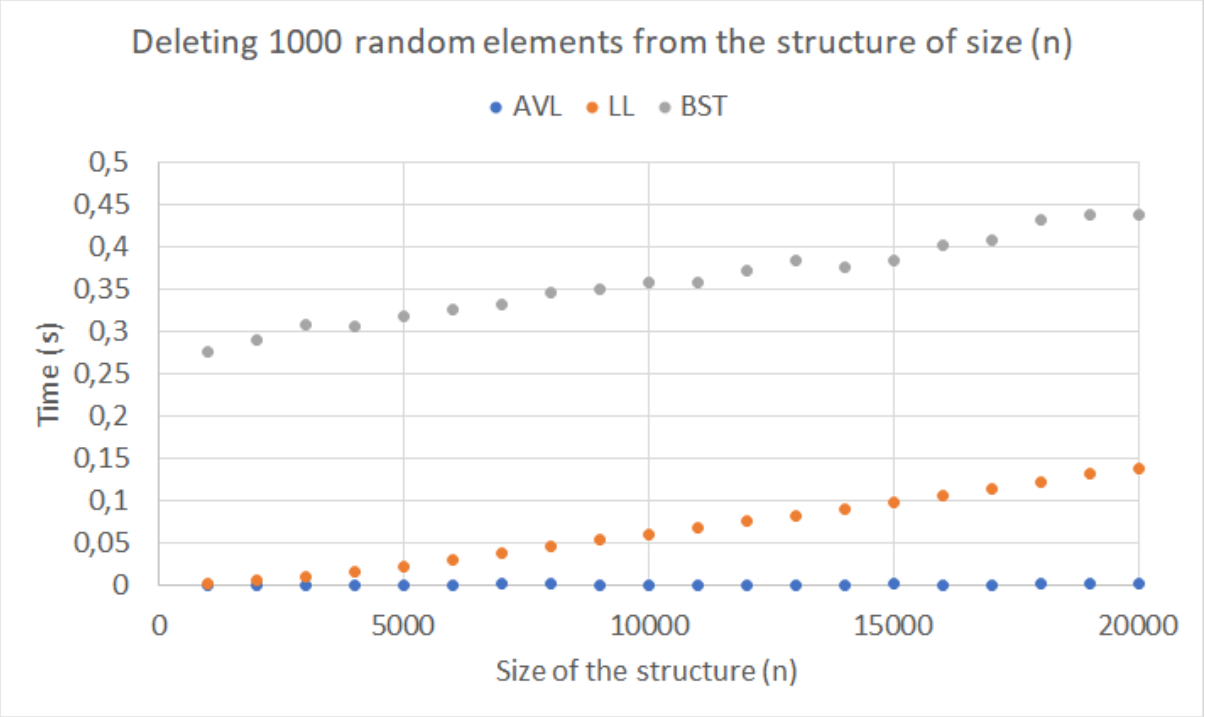

This graph also proves that operation of balancing the AVL tree takes some time, therefore BST is faster in this case. Nethertheless both of the trees are incomparably faster than the linked list in which deleting 100 000 elements in random order lasted 40,329 seconds (average of 10 runs).

## Worst case scenario.

The biggest disadvantage of standard BST is the possibility of transforming into a Linked List. We ran a test in which we created a database with ordered indices. Here are some measurements from this test:






***Expectations:***

AVL should perform great and stay at it's standard time complexity $O(\log_2(n))$. The Linked List should also stay at the same complexity, but operations should last longer thus every insertion of the element would have to iterate to the end of the current list. BST should become linked list i.e. change it's complexity to O(n).

***Observations and conclusions:***

What can be easily noticed, is the fact that BST started to behave dramatically worse. We expected it to be at the similar level as a linked list, but to our surprise graphs showed something different. After some investigation and checking our code, we came up with a conclusion. In our opinion the reason for such a behavior of BST is caused by a recursive nature of trees thus our functions (adding, searching and deleting) are recursive in contrast to the linked list, whose functions are implemented iteratively.

Also, as expected AVL outclassed BST and LL in such a case scenario thanks to the ability of balancing itself.

**Final conclusions:**
After testing and implementing all three data structures as expected, AVL turns out to be the safest data structure for storing data, but it's also the hardest to understand and implement. Nevertheless BST performs very similarly, and is noticeably easier to understand and implement. Worst case of inserting data is very unlikely and could be easily solved by simply shuffling the database. Last but not least is Ordered Linked List. This structure is definitely the easiest to understand and implement. It works acceptably well for small data instances, and can be found useful in many programs. (As a fun fact we can say that we've used this structure in our program for Low Level Programming laboratories.)
Below are presented average time complexities of tested structures, which were confirmed by our test.

| | Addiction | Searching | Removing |
|---|---|---|---|
| Ordered linked list | n | n | n |
| Binary search tree | log n | log n | log n |
| Balanced binary search tree | log n | log n | log n |

***Below are all tables that we've created. Let's call it resources. In most cases time was measured every 10 000 elements i.e. after adding 10 000 elements timer was stopped, search started, after searching all elements added so far adding was resumed until adding next 10 000 elements, and so on.***

1. Table with times related to addition of elements to structures.

| LL iterator | Trees iterator | BST | BST prefix sum | Linked list | Linked list prefix | AVL | AVL prefix sum | AVL adding one element | BST adding one element | LL adding one element |
|---|---|---|---|---|---|---|---|---|---|---|
| 1000 | 10000 | 0,0087 | 0,0087 | 0,0009 | 0,0009 | 0,0097 | 0,0097 | 0,00000097 | 0,00000087 | 0,0000009 |
| 2000 | 20000 | 0,0098 | 0,0185 | 0,0044 | 0,0053 | 0,0129 | 0,0226 | 0,00000129 | 0,00000098 | 0,0000044 |
| 3000 | 30000 | 0,0134 | 0,0319 | 0,0094 | 0,0147 | 0,0155 | 0,0381 | 0,00000155 | 0,00000134 | 0,0000094 |
| 4000 | 40000 | 0,0147 | 0,0466 | 0,0104 | 0,0251 | 0,0157 | 0,0538 | 0,00000157 | 0,00000147 | 0,0000104 |
| 5000 | 50000 | 0,0188 | 0,0654 | 0,0144 | 0,0395 | 0,0126 | 0,0664 | 0,00000126 | 0,00000188 | 0,0000144 |
| 6000 | 60000 | 0,0195 | 0,0849 | 0,0201 | 0,0596 | 0,0122 | 0,0786 | 0,00000122 | 0,00000195 | 0,0000201 |
| 7000 | 70000 | 0,019 | 0,1039 | 0,0261 | 0,0857 | 0,0126 | 0,0912 | 0,00000126 | 0,0000019 | 0,0000261 |
| 8000 | 80000 | 0,0159 | 0,1198 | 0,0348 | 0,1205 | 0,0131 | 0,1043 | 0,00000131 | 0,00000159 | 0,0000348 |
| 9000 | 90000 | 0,0117 | 0,1315 | 0,0417 | 0,1622 | 0,0137 | 0,118 | 0,00000137 | 0,00000117 | 0,0000417 |
| 10000 | 100000 | 0,0124 | 0,1439 | 0,0485 | 0,2107 | 0,0142 | 0,1322 | 0,00000142 | 0,00000124 | 0,0000485 |
| 11000 | 110000 | 0,0131 | 0,157 | 0,056 | 0,2667 | 0,0132 | 0,1454 | 0,00000132 | 0,00000131 | 0,000056 |
| 12000 | 120000 | 0,0112 | 0,1682 | 0,0633 | 0,33 | 0,0139 | 0,1593 | 0,00000139 | 0,00000112 | 0,0000633 |
| 13000 | 130000 | 0,0126 | 0,1808 | 0,0719 | 0,4019 | 0,014 | 0,1733 | 0,0000014 | 0,00000126 | 0,0000719 |
| 14000 | 140000 | 0,0121 | 0,1929 | 0,0768 | 0,4787 | 0,0141 | 0,1874 | 0,00000141 | 0,00000121 | 0,0000768 |
| 15000 | 150000 | 0,0122 | 0,2051 | 0,0858 | 0,5645 | 0,0137 | 0,2011 | 0,00000137 | 0,00000122 | 0,0000858 |
| 16000 | 160000 | 0,0125 | 0,2176 | 0,0901 | 0,6546 | 0,0138 | 0,2149 | 0,00000138 | 0,00000125 | 0,0000901 |
| 17000 | 170000 | 0,0131 | 0,2307 | 0,0973 | 0,7519 | 0,0143 | 0,2292 | 0,00000143 | 0,00000131 | 0,0000973 |
| 18000 | 180000 | 0,0129 | 0,2436 | 0,1049 | 0,8568 | 0,0142 | 0,2434 | 0,00000142 | 0,00000129 | 0,0001049 |
| 19000 | 190000 | 0,0126 | 0,2562 | 0,1118 | 0,9686 | 0,0143 | 0,2577 | 0,00000143 | 0,00000126 | 0,0001118 |
| 20000 | 200000 | 0,0128 | 0,269 | 0,1186 | 1,0872 | 0,0145 | 0,2722 | 0,00000145 | 0,00000128 | 0,0001186 |
| 21000 | 210000 | 0,0133 | 0,2823 | 0,1243 | 1,2115 | 0,015 | 0,2872 | 0,0000015 | 0,00000133 | 0,0001243 |
| 22000 | 220000 | 0,0128 | 0,2951 | 0,1334 | 1,3449 | 0,0147 | 0,3019 | 0,00000147 | 0,00000128 | 0,0001334 |
| 23000 | 230000 | 0,0131 | 0,3082 | 0,1384 | 1,4833 | 0,0153 | 0,3172 | 0,00000153 | 0,00000131 | 0,0001384 |
| 24000 | 240000 | 0,013 | 0,3212 | 0,148 | 1,6313 | 0,0157 | 0,3329 | 0,00000157 | 0,0000013 | 0,000148 |
| 25000 | 250000 | 0,0136 | 0,3348 | 0,1544 | 1,7857 | 0,0151 | 0,348 | 0,00000151 | 0,00000136 | 0,0001544 |
| 26000 | 260000 | 0,0147 | 0,3495 | 0,1611 | 1,9468 | 0,0152 | 0,3632 | 0,00000152 | 0,00000147 | 0,0001611 |
| 27000 | 270000 | 0,0143 | 0,3638 | 0,1671 | 2,1139 | 0,0151 | 0,3783 | 0,00000151 | 0,00000143 | 0,0001671 |
| 28000 | 280000 | 0,0153 | 0,3791 | 0,1746 | 2,2885 | 0,0154 | 0,3937 | 0,00000154 | 0,00000153 | 0,0001746 |
| 29000 | 290000 | 0,015 | 0,3941 | 0,1843 | 2,4728 | 0,0159 | 0,4096 | 0,00000159 | 0,0000015 | 0,0001843 |
| 30000 | 300000 | 0,0142 | 0,4083 | 0,1878 | 2,6606 | 0,0154 | 0,425 | 0,00000154 | 0,00000142 | 0,0001878 |
| 31000 | 310000 | 0,0133 | 0,4216 | 0,196 | 2,8566 | 0,0156 | 0,4406 | 0,00000156 | 0,00000133 | 0,000196 |
| 32000 | 320000 | 0,0134 | 0,435 | 0,2026 | 3,0592 | 0,0162 | 0,4568 | 0,00000162 | 0,00000134 | 0,0002026 |
| 33000 | 330000 | 0,0135 | 0,4485 | 0,2107 | 3,2699 | 0,0166 | 0,4734 | 0,00000166 | 0,00000135 | 0,0002107 |
| 34000 | 340000 | 0,0149 | 0,4634 | 0,2185 | 3,4884 | 0,0168 | 0,4902 | 0,00000168 | 0,00000149 | 0,0002185 |
| 35000 | 350000 | 0,0146 | 0,478 | 0,2242 | 3,7126 | 0,0164 | 0,5066 | 0,00000164 | 0,00000146 | 0,0002242 |
| 36000 | 360000 | 0,0155 | 0,4935 | 0,2313 | 3,9439 | 0,0163 | 0,5229 | 0,00000163 | 0,00000155 | 0,0002313 |
| 37000 | 370000 | 0,0143 | 0,5078 | 0,2419 | 4,1858 | 0,0162 | 0,5391 | 0,00000162 | 0,00000143 | 0,0002419 |
| 38000 | 380000 | 0,0146 | 0,5224 | 0,2487 | 4,4345 | 0,017 | 0,5561 | 0,0000017 | 0,00000146 | 0,0002487 |
| 39000 | 390000 | 0,0162 | 0,5386 | 0,2563 | 4,6908 | 0,0166 | 0,5727 | 0,00000166 | 0,00000162 | 0,0002563 |
| 40000 | 400000 | 0,0139 | 0,5525 | 0,2647 | 4,9555 | 0,018 | 0,5907 | 0,0000018 | 0,00000139 | 0,0002647 |
| 41000 | 410000 | 0,0163 | 0,5688 | 0,2731 | 5,2286 | 0,0171 | 0,6078 | 0,00000171 | 0,00000163 | 0,0002731 |
| 42000 | 420000 | 0,0144 | 0,5832 | 0,2735 | 5,5021 | 0,0174 | 0,6252 | 0,00000174 | 0,00000144 | 0,0002735 |
| 43000 | 430000 | 0,0147 | 0,5979 | 0,2853 | 5,7874 | 0,0172 | 0,6424 | 0,00000172 | 0,00000147 | 0,0002853 |
| 44000 | 440000 | 0,0149 | 0,6128 | 0,2881 | 6,0755 | 0,0175 | 0,6599 | 0,00000175 | 0,00000149 | 0,0002881 |
| 45000 | 450000 | 0,016 | 0,6288 | 0,2934 | 6,3689 | 0,0168 | 0,6767 | 0,00000168 | 0,0000016 | 0,0002934 |
| 46000 | 460000 | 0,0153 | 0,6441 | 0,3059 | 6,6748 | 0,0169 | 0,6936 | 0,00000169 | 0,00000153 | 0,0003059 |
| 47000 | 470000 | 0,0249 | 0,669 | 0,3217 | 6,9965 | 0,0179 | 0,7115 | 0,00000179 | 0,00000249 | 0,0003217 |
| 48000 | 480000 | 0,0151 | 0,6841 | 0,3208 | 7,3173 | 0,018 | 0,7295 | 0,0000018 | 0,00000151 | 0,0003208 |
| 49000 | 490000 | 0,0154 | 0,6995 | 0,3305 | 7,6478 | 0,0178 | 0,7473 | 0,00000178 | 0,00000154 | 0,0003305 |
| 50000 | 500000 | 0,0147 | 0,7142 | 0,3314 | 7,9792 | 0,0177 | 0,765 | 0,00000177 | 0,00000147 | 0,0003314 |
| 51000 | 510000 | 0,0148 | 0,729 | 0,3465 | 8,3257 | 0,0178 | 0,7828 | 0,00000178 | 0,00000148 | 0,0003465 |
| 52000 | 520000 | 0,0145 | 0,7435 | 0,3541 | 8,6798 | 0,0176 | 0,8004 | 0,00000176 | 0,00000145 | 0,0003541 |
| 53000 | 530000 | 0,0152 | 0,7587 | 0,3592 | 9,039 | 0,0175 | 0,8179 | 0,00000175 | 0,00000152 | 0,0003592 |
| 54000 | 540000 | 0,0147 | 0,7734 | 0,3569 | 9,3959 | 0,0176 | 0,8355 | 0,00000176 | 0,00000147 | 0,0003569 |
| 55000 | 550000 | 0,0156 | 0,789 | 0,3681 | 9,764 | 0,0183 | 0,8538 | 0,00000183 | 0,00000156 | 0,0003681 |
| 56000 | 560000 | 0,0165 | 0,8055 | 0,3793 | 10,1433 | 0,0175 | 0,8713 | 0,00000175 | 0,00000165 | 0,0003793 |
| 57000 | 570000 | 0,0151 | 0,8206 | 0,3871 | 10,5304 | 0,018 | 0,8893 | 0,0000018 | 0,00000151 | 0,0003871 |
| 58000 | 580000 | 0,0152 | 0,8358 | 0,3919 | 10,9223 | 0,0179 | 0,9072 | 0,00000179 | 0,00000152 | 0,0003919 |
| 59000 | 590000 | 0,0156 | 0,8514 | 0,4019 | 11,3242 | 0,018 | 0,9252 | 0,0000018 | 0,00000156 | 0,0004019 |
| 60000 | 600000 | 0,0163 | 0,8677 | 0,4071 | 11,7313 | 0,018 | 0,9432 | 0,0000018 | 0,00000163 | 0,0004071 |
| 61000 | 610000 | 0,0174 | 0,8851 | 0,4222 | 12,1535 | 0,018 | 0,9612 | 0,0000018 | 0,00000174 | 0,0004222 |
| 62000 | 620000 | 0,0169 | 0,902 | 0,4253 | 12,5788 | 0,018 | 0,9792 | 0,0000018 | 0,00000169 | 0,0004253 |
| 63000 | 630000 | 0,0158 | 0,9178 | 0,4298 | 13,0086 | 0,0182 | 0,9974 | 0,00000182 | 0,00000158 | 0,0004298 |
| 64000 | 640000 | 0,0167 | 0,9345 | 0,4381 | 13,4467 | 0,0181 | 1,0155 | 0,00000181 | 0,00000167 | 0,0004381 |
| 65000 | 650000 | 0,0168 | 0,9513 | 0,4509 | 13,8976 | 0,0182 | 1,0337 | 0,00000182 | 0,00000168 | 0,0004509 |
| 66000 | 660000 | 0,0159 | 0,9672 | 0,4536 | 14,3512 | 0,018 | 1,0517 | 0,0000018 | 0,00000159 | 0,0004536 |
| 67000 | 670000 | 0,0164 | 0,9836 | 0,465 | 14,8162 | 0,0183 | 1,07 | 0,00000183 | 0,00000164 | 0,000465 |
| 68000 | 680000 | 0,0158 | 0,9994 | 0,467 | 15,2832 | 0,0187 | 1,0887 | 0,00000187 | 0,00000158 | 0,000467 |
| 69000 | 690000 | 0,0162 | 1,0156 | 0,4796 | 15,7628 | 0,0184 | 1,1071 | 0,00000184 | 0,00000162 | 0,0004796 |
| 70000 | 700000 | 0,0178 | 1,0334 | 0,4947 | 16,2575 | 0,019 | 1,1261 | 0,0000019 | 0,00000178 | 0,0004947 |
| 71000 | 710000 | 0,0168 | 1,0502 | 0,4981 | 16,7556 | 0,0184 | 1,1445 | 0,00000184 | 0,00000168 | 0,0004981 |
| 72000 | 720000 | 0,0162 | 1,0664 | 0,5129 | 17,2685 | 0,0227 | 1,1672 | 0,00000227 | 0,00000162 | 0,0005129 |
| 73000 | 730000 | 0,0148 | 1,0812 | 0,5245 | 17,793 | 0,0187 | 1,1859 | 0,00000187 | 0,00000148 | 0,0005245 |
| 74000 | 740000 | 0,0161 | 1,0973 | 0,5411 | 18,3341 | 0,0198 | 1,2057 | 0,00000198 | 0,00000161 | 0,0005411 |
| 75000 | 750000 | 0,0168 | 1,1141 | 0,549 | 18,8831 | 0,0199 | 1,2256 | 0,00000199 | 0,00000168 | 0,000549 |
| 76000 | 760000 | 0,0181 | 1,1322 | 0,5602 | 19,4433 | 0,0188 | 1,2444 | 0,00000188 | 0,00000181 | 0,0005602 |
| 77000 | 770000 | 0,0165 | 1,1487 | 0,5593 | 20,0026 | 0,0188 | 1,2632 | 0,00000188 | 0,00000165 | 0,0005593 |
| 78000 | 780000 | 0,0163 | 1,165 | 0,5683 | 20,5709 | 0,0192 | 1,2824 | 0,00000192 | 0,00000163 | 0,0005683 |
| 79000 | 790000 | 0,0176 | 1,1826 | 0,6403 | 21,2112 | 0,019 | 1,3014 | 0,0000019 | 0,00000176 | 0,0006403 |
| 80000 | 800000 | 0,0246 | 1,2072 | 0,5908 | 21,802 | 0,0186 | 1,32 | 0,00000186 | 0,00000246 | 0,0005908 |
| 81000 | 810000 | 0,0174 | 1,2246 | 0,602 | 22,404 | 0,0203 | 1,3403 | 0,00000203 | 0,00000174 | 0,000602 |
| 82000 | 820000 | 0,0229 | 1,2475 | 0,6208 | 23,0248 | 0,0188 | 1,3591 | 0,00000188 | 0,00000229 | 0,0006208 |
| 83000 | 830000 | 0,0164 | 1,2639 | 0,6287 | 23,6535 | 0,0193 | 1,3784 | 0,00000193 | 0,00000164 | 0,0006287 |
| 84000 | 840000 | 0,0202 | 1,2841 | 0,6395 | 24,293 | 0,0193 | 1,3977 | 0,00000193 | 0,00000202 | 0,0006395 |
| 85000 | 850000 | 0,0177 | 1,3018 | 0,6529 | 24,9459 | 0,019 | 1,4167 | 0,0000019 | 0,00000177 | 0,0006529 |
| 86000 | 860000 | 0,0174 | 1,3192 | 0,6573 | 25,6032 | 0,0187 | 1,4354 | 0,00000187 | 0,00000174 | 0,0006573 |
| 87000 | 870000 | 0,0188 | 1,338 | 0,6662 | 26,2694 | 0,0191 | 1,4545 | 0,00000191 | 0,00000188 | 0,0006662 |
| 88000 | 880000 | 0,017 | 1,355 | 0,6865 | 26,9559 | 0,0193 | 1,4738 | 0,00000193 | 0,0000017 | 0,0006865 |
| 89000 | 890000 | 0,0163 | 1,3713 | 0,6949 | 27,6508 | 0,0194 | 1,4932 | 0,00000194 | 0,00000163 | 0,0006949 |
| 90000 | 900000 | 0,0173 | 1,3886 | 0,7223 | 28,3731 | 0,0191 | 1,5123 | 0,00000191 | 0,00000173 | 0,0007223 |
| 91000 | 910000 | 0,0189 | 1,4075 | 0,7402 | 29,1133 | 0,0195 | 1,5318 | 0,00000195 | 0,00000189 | 0,0007402 |
| 92000 | 920000 | 0,017 | 1,4245 | 0,7337 | 29,847 | 0,0196 | 1,5514 | 0,00000196 | 0,0000017 | 0,0007337 |
| 93000 | 930000 | 0,0169 | 1,4414 | 0,7593 | 30,6063 | 0,019 | 1,5704 | 0,0000019 | 0,00000169 | 0,0007593 |
| 94000 | 940000 | 0,0158 | 1,4572 | 0,7701 | 31,3764 | 0,0196 | 1,59 | 0,00000196 | 0,00000158 | 0,0007701 |
| 95000 | 950000 | 0,0168 | 1,474 | 0,7953 | 32,1717 | 0,0193 | 1,6093 | 0,00000193 | 0,00000168 | 0,0007953 |
| 96000 | 960000 | 0,0218 | 1,4958 | 0,8104 | 32,9821 | 0,0196 | 1,6289 | 0,00000196 | 0,00000218 | 0,0008104 |
| 97000 | 970000 | 0,0222 | 1,518 | 0,8165 | 33,7986 | 0,0194 | 1,6483 | 0,00000194 | 0,00000222 | 0,0008165 |
| 98000 | 980000 | 0,0169 | 1,5349 | 0,8565 | 34,6551 | 0,0228 | 1,6711 | 0,00000228 | 0,00000169 | 0,0008565 |
| 99000 | 990000 | 0,0185 | 1,5534 | 0,8612 | 35,5163 | 0,0194 | 1,6905 | 0,00000194 | 0,00000185 | 0,0008612 |
| 100000 | 1000000 | 0,0169 | 1,5703 | 0,8726 | 36,3889 | 0,0198 | 1,7103 | 0,00000198 | 0,00000169 | 0,0008726 |

2. Times related to searching of elements in structures:

| Trees iterator | LL iterator | AVL | AVL time per one element | LL | LL time per one element | BST | BST time per one element |
|---|---|---|---|---|---|---|---|
| 10000 | 1000 | 0,0015 | 1,50E-07 | 0,0026 | 0,00000026 | 0,0016 | 0,00000016 |
| 20000 | 2000 | 0,0037 | 1,85E-07 | 0,0062 | 0,00000031 | 0,004 | 0,0000002 |
| 30000 | 3000 | 0,0063 | 2,10E-07 | 0,0103 | 3,43333E-07 | 0,0069 | 0,00000023 |
| 40000 | 4000 | 0,0093 | 2,33E-07 | 0,0132 | 0,00000033 | 0,0116 | 0,00000029 |
| 50000 | 5000 | 0,0131 | 2,62E-07 | 0,0177 | 0,000000354 | 0,0137 | 0,000000274 |
| 60000 | 6000 | 0,0176 | 2,93E-07 | 0,0244 | 4,06667E-07 | 0,0165 | 0,000000275 |
| 70000 | 7000 | 0,019 | 2,71E-07 | 0,0308 | 0,00000044 | 0,021 | 0,0000003 |
| 80000 | 8000 | 0,021 | 2,63E-07 | 0,0402 | 5,025E-07 | 0,0263 | 3,2875E-07 |
| 90000 | 9000 | 0,0284 | 3,16E-07 | 0,0492 | 5,46667E-07 | 0,03 | 3,33333E-07 |
| 100000 | 10000 | 0,0348 | 3,48E-07 | 0,0566 | 0,000000566 | 0,0359 | 0,000000359 |
| 110000 | 11000 | 0,036 | 3,27E-07 | 0,0648 | 5,89091E-07 | 0,0397 | 3,60909E-07 |
| 120000 | 12000 | 0,0417 | 3,48E-07 | 0,0713 | 5,94167E-07 | 0,0469 | 3,90833E-07 |
| 130000 | 13000 | 0,0444 | 3,42E-07 | 0,0802 | 6,16923E-07 | 0,0489 | 3,76154E-07 |
| 140000 | 14000 | 0,0506 | 3,61E-07 | 0,0866 | 6,18571E-07 | 0,0588 | 0,00000042 |
| 150000 | 15000 | 0,0516 | 3,44E-07 | 0,0938 | 6,25333E-07 | 0,0602 | 4,01333E-07 |
| 160000 | 16000 | 0,0552 | 3,45E-07 | 0,1 | 0,000000625 | 0,0677 | 4,23125E-07 |
| 170000 | 17000 | 0,062 | 3,65E-07 | 0,1099 | 6,46471E-07 | 0,0793 | 4,66471E-07 |
| 180000 | 18000 | 0,0715 | 3,97E-07 | 0,1165 | 6,47222E-07 | 0,081 | 0,00000045 |
| 190000 | 19000 | 0,079 | 4,16E-07 | 0,1241 | 6,53158E-07 | 0,0867 | 4,56316E-07 |
| 200000 | 20000 | 0,0858 | 4,29E-07 | 0,1311 | 6,555E-07 | 0,0918 | 0,000000459 |
| 210000 | 21000 | 0,0883 | 4,20E-07 | 0,1372 | 6,53333E-07 | 0,0973 | 4,63333E-07 |
| 220000 | 22000 | 0,0909 | 4,13E-07 | 0,1461 | 6,64091E-07 | 0,1037 | 4,71364E-07 |
| 230000 | 23000 | 0,0953 | 4,14E-07 | 0,156 | 6,78261E-07 | 0,1146 | 4,98261E-07 |
| 240000 | 24000 | 0,1002 | 4,18E-07 | 0,1645 | 6,85417E-07 | 0,1199 | 4,99583E-07 |
| 250000 | 25000 | 0,1085 | 4,34E-07 | 0,1724 | 6,896E-07 | 0,127 | 0,000000508 |
| 260000 | 26000 | 0,1195 | 4,60E-07 | 0,1759 | 6,76538E-07 | 0,1395 | 5,36538E-07 |
| 270000 | 27000 | 0,1233 | 4,57E-07 | 0,1853 | 6,86296E-07 | 0,1397 | 5,17407E-07 |
| 280000 | 28000 | 0,1356 | 4,84E-07 | 0,2196 | 7,84286E-07 | 0,1446 | 5,16429E-07 |
| 290000 | 29000 | 0,1347 | 4,64E-07 | 0,2035 | 7,01724E-07 | 0,1521 | 5,24483E-07 |
| 300000 | 30000 | 0,1402 | 4,67E-07 | 0,2066 | 6,88667E-07 | 0,1641 | 0,000000547 |
| 310000 | 31000 | 0,1465 | 4,73E-07 | 0,2162 | 6,97419E-07 | 0,1718 | 5,54194E-07 |
| 320000 | 32000 | 0,1536 | 4,80E-07 | 0,2266 | 7,08125E-07 | 0,1735 | 5,42188E-07 |
| 330000 | 33000 | 0,1713 | 5,19E-07 | 0,2328 | 7,05455E-07 | 0,1878 | 5,69091E-07 |
| 340000 | 34000 | 0,1825 | 5,37E-07 | 0,2407 | 7,07941E-07 | 0,1887 | 0,000000555 |
| 350000 | 35000 | 0,1877 | 5,36E-07 | 0,2467 | 7,04857E-07 | 0,2066 | 5,90286E-07 |
| 360000 | 36000 | 0,1826 | 5,07E-07 | 0,2583 | 7,175E-07 | 0,2162 | 6,00556E-07 |
| 370000 | 37000 | 0,1909 | 5,16E-07 | 0,2716 | 7,34054E-07 | 0,2162 | 5,84324E-07 |
| 380000 | 38000 | 0,1982 | 5,22E-07 | 0,2746 | 7,22632E-07 | 0,2242 | 0,00000059 |
| 390000 | 39000 | 0,1963 | 5,03E-07 | 0,2812 | 7,21026E-07 | 0,2302 | 5,90256E-07 |
| 400000 | 40000 | 0,2052 | 5,13E-07 | 0,2879 | 7,1975E-07 | 0,2463 | 6,1575E-07 |
| 410000 | 41000 | 0,2046 | 4,99E-07 | 0,2975 | 7,2561E-07 | 0,2463 | 6,00732E-07 |
| 420000 | 42000 | 0,2279 | 5,43E-07 | 0,3 | 7,14286E-07 | 0,2572 | 6,12381E-07 |
| 430000 | 43000 | 0,2277 | 5,30E-07 | 0,3169 | 7,36977E-07 | 0,2662 | 6,1907E-07 |
| 540000 | 54000 | 0,3004 | 5,56E-07 | 0,4068 | 7,53333E-07 | 0,3523 | 6,52407E-07 |
| 550000 | 55000 | 0,3156 | 5,74E-07 | 0,4164 | 7,57091E-07 | 0,3707 | 0,000000674 |
| 560000 | 56000 | 0,3137 | 5,60E-07 | 0,4103 | 7,32679E-07 | 0,3708 | 6,62143E-07 |
| 570000 | 57000 | 0,3241 | 5,69E-07 | 0,4259 | 7,47193E-07 | 0,3766 | 6,60702E-07 |
| 580000 | 58000 | 0,3383 | 5,83E-07 | 0,4397 | 7,58103E-07 | 0,3911 | 6,7431E-07 |
| 590000 | 59000 | 0,3369 | 5,71E-07 | 0,4452 | 7,54576E-07 | 0,3925 | 6,65254E-07 |
| 600000 | 60000 | 0,3425 | 5,71E-07 | 0,4465 | 7,44167E-07 | 0,414 | 0,00000069 |
| 610000 | 61000 | 0,3521 | 5,77E-07 | 0,4563 | 7,48033E-07 | 0,4152 | 6,80656E-07 |
| 620000 | 62000 | 0,3529 | 5,69E-07 | 0,4643 | 7,48871E-07 | 0,4356 | 7,02581E-07 |
| 630000 | 63000 | 0,3713 | 5,89E-07 | 0,4979 | 7,90317E-07 | 0,4302 | 6,82857E-07 |
| 640000 | 64000 | 0,3683 | 5,75E-07 | 0,4949 | 7,73281E-07 | 0,4479 | 6,99844E-07 |
| 650000 | 65000 | 0,3843 | 5,91E-07 | 0,5035 | 7,74615E-07 | 0,4505 | 6,93077E-07 |
| 660000 | 66000 | 0,3914 | 5,93E-07 | 0,5772 | 8,74545E-07 | 0,4622 | 7,00303E-07 |
| 670000 | 67000 | 0,4046 | 6,04E-07 | 0,5157 | 7,69701E-07 | 0,4699 | 7,01343E-07 |
| 680000 | 68000 | 0,4124 | 6,06E-07 | 0,5268 | 7,74706E-07 | 0,482 | 7,08824E-07 |
| 690000 | 69000 | 0,4086 | 5,92E-07 | 0,5424 | 7,86087E-07 | 0,4907 | 7,11159E-07 |
| 700000 | 70000 | 0,4228 | 6,04E-07 | 0,5555 | 7,93571E-07 | 0,5029 | 7,18429E-07 |
| 710000 | 71000 | 0,4154 | 5,85E-07 | 0,5615 | 7,90845E-07 | 0,5077 | 7,1507E-07 |
| 720000 | 72000 | 0,4363 | 6,06E-07 | 0,5693 | 7,90694E-07 | 0,5312 | 7,37778E-07 |
| 730000 | 73000 | 0,4397 | 6,02E-07 | 0,5693 | 7,79863E-07 | 0,5372 | 7,3589E-07 |
| 740000 | 74000 | 0,4523 | 6,11E-07 | 0,5764 | 7,78919E-07 | 0,5409 | 7,30946E-07 |
| 750000 | 75000 | 0,4544 | 6,06E-07 | 0,6081 | 8,108E-07 | 0,5426 | 7,23467E-07 |
| 760000 | 76000 | 0,464 | 6,11E-07 | 0,6084 | 8,00526E-07 | 0,5572 | 7,33158E-07 |
| 770000 | 77000 | 0,4742 | 6,16E-07 | 0,619 | 8,03896E-07 | 0,5652 | 7,34026E-07 |
| 780000 | 78000 | 0,4807 | 6,16E-07 | 0,6275 | 8,04487E-07 | 0,5709 | 7,31923E-07 |
| 790000 | 79000 | 0,4869 | 6,16E-07 | 0,6295 | 7,96835E-07 | 0,5871 | 7,43165E-07 |
| 800000 | 80000 | 0,4953 | 6,19E-07 | 0,6547 | 8,18375E-07 | 0,5924 | 7,405E-07 |
| 810000 | 81000 | 0,4983 | 6,15E-07 | 0,6607 | 8,15679E-07 | 0,6014 | 7,42469E-07 |
| 820000 | 82000 | 0,5133 | 6,26E-07 | 0,6657 | 8,11829E-07 | 0,6151 | 7,50122E-07 |
| 830000 | 83000 | 0,51 | 6,14E-07 | 0,6761 | 8,14578E-07 | 0,6143 | 7,4012E-07 |
| 840000 | 84000 | 0,533 | 6,35E-07 | 0,6986 | 8,31667E-07 | 0,6466 | 7,69762E-07 |
| 850000 | 85000 | 0,5492 | 6,46E-07 | 0,6981 | 8,21294E-07 | 0,6512 | 7,66118E-07 |
| 860000 | 86000 | 0,5666 | 6,59E-07 | 0,7178 | 8,34651E-07 | 0,6543 | 7,60814E-07 |
| 870000 | 87000 | 0,5885 | 6,76E-07 | 0,7152 | 8,22069E-07 | 0,6625 | 7,61494E-07 |
| 880000 | 88000 | 0,6084 | 6,91E-07 | 0,7452 | 8,46818E-07 | 0,672 | 7,63636E-07 |
| 990000 | 99000 | 0,6604 | 6,67E-07 | 0,9255 | 9,34848E-07 | 0,7912 | 7,99192E-07 |
| 1000000 | 100000 | 0,6634 | 6,63E-07 | 0,9443 | 9,443E-07 | 0,7965 | 7,965E-07 |

3. Table related to times of deleting elements from structures:

| iterator | Deleting 10 000 elements from LL | Deleting 10 000 elements from BST | Deleting 10 000 elements from AVL | LL / iterator | BST/iterator | AVL/iterator |
|---|---|---|---|---|---|---|
| 10000 | 0,2849 | 0,0122 | 0,0138 | 0,00002849 | 1,22E-06 | 1,38E-06 |
| 20000 | 1,0236 | 0,0043 | 0,0061 | 0,00005118 | 2,15E-07 | 3,05E-07 |
| 30000 | 1,7748 | 0,0048 | 0,0069 | 0,00005916 | 1,60E-07 | 2,30E-07 |
| 40000 | 2,5664 | 0,0047 | 0,0079 | 0,00006416 | 1,18E-07 | 1,98E-07 |
| 50000 | 3,3661 | 0,0056 | 0,0078 | 0,000067322 | 1,12E-07 | 1,56E-07 |
| 60000 | 4,2348 | 0,0056 | 0,0087 | 0,00007058 | 9,33E-08 | 1,45E-07 |
| 70000 | 5,1551 | 0,0058 | 0,009 | 7,36443E-05 | 8,29E-08 | 1,29E-07 |
| 80000 | 6,1977 | 0,0058 | 0,0105 | 7,74713E-05 | 7,25E-08 | 1,31E-07 |
| 90000 | 7,2694 | 0,0059 | 0,011 | 8,07711E-05 | 6,56E-08 | 1,22E-07 |
| 100000 | 8,4562 | 0,0062 | 0,0109 | 0,000084562 | 6,20E-08 | 1,09E-07 |
| 110000 | | 0,0062 | 0,0101 | | 5,64E-08 | 9,18E-08 |
| 120000 | | 0,0073 | 0,0101 | | 6,08E-08 | 8,42E-08 |
| 130000 | | 0,0072 | 0,0104 | | 5,54E-08 | 8,00E-08 |
| 140000 | | 0,0074 | 0,0109 | | 5,29E-08 | 7,79E-08 |
| 150000 | | 0,0076 | 0,0114 | | 5,07E-08 | 7,60E-08 |
| 160000 | | 0,0077 | 0,0118 | | 4,81E-08 | 7,38E-08 |
| 170000 | | 0,0076 | 0,011 | | 4,47E-08 | 6,47E-08 |
| 180000 | | 0,0079 | 0,0114 | | 4,39E-08 | 6,33E-08 |
| 190000 | | 0,0081 | 0,0121 | | 4,26E-08 | 6,37E-08 |
| 200000 | | 0,0085 | 0,0125 | | 4,25E-08 | 6,25E-08 |
| 210000 | | 0,0082 | 0,0118 | | 3,90E-08 | 5,62E-08 |
| 220000 | | 0,0078 | 0,0113 | | 3,55E-08 | 5,14E-08 |
| 230000 | | 0,0083 | 0,0116 | | 3,61E-08 | 5,04E-08 |
| 240000 | | 0,0086 | 0,0121 | | 3,58E-08 | 5,04E-08 |
| 250000 | | 0,0088 | 0,0118 | | 3,52E-08 | 4,72E-08 |
| 260000 | | 0,0088 | 0,0124 | | 3,38E-08 | 4,77E-08 |
| 270000 | | 0,0095 | 0,0136 | | 3,52E-08 | 5,04E-08 |
| 280000 | | 0,0091 | 0,0135 | | 3,25E-08 | 4,82E-08 |
| 290000 | | 0,0095 | 0,0128 | | 3,28E-08 | 4,41E-08 |
| 300000 | | 0,0089 | 0,0125 | | 2,97E-08 | 4,17E-08 |
| 310000 | | 0,0087 | 0,0133 | | 2,81E-08 | 4,29E-08 |
| 320000 | | 0,0083 | 0,0135 | | 2,59E-08 | 4,22E-08 |
| 330000 | | 0,0083 | 0,0132 | | 2,52E-08 | 4,00E-08 |
| 340000 | | 0,0087 | 0,0132 | | 2,56E-08 | 3,88E-08 |
| 350000 | | 0,0086 | 0,0128 | | 2,46E-08 | 3,66E-08 |
| 360000 | | 0,0092 | 0,0129 | | 2,56E-08 | 3,58E-08 |
| 370000 | | 0,0091 | 0,0126 | | 2,46E-08 | 3,41E-08 |
| 380000 | | 0,01 | 0,0128 | | 2,63E-08 | 3,37E-08 |
| 390000 | | 0,0096 | 0,0131 | | 2,46E-08 | 3,36E-08 |
| 400000 | | 0,01 | 0,0131 | | 2,50E-08 | 3,28E-08 |
| 410000 | | 0,0102 | 0,0132 | | 2,49E-08 | 3,22E-08 |
| 420000 | | 0,0097 | 0,0128 | | 2,31E-08 | 3,05E-08 |
| 430000 | | 0,0095 | 0,0131 | | 2,21E-08 | 3,05E-08 |
| 440000 | | 0,0091 | 0,0133 | | 2,07E-08 | 3,02E-08 |
| 450000 | | 0,0094 | 0,0138 | | 2,09E-08 | 3,07E-08 |
| 460000 | | 0,0096 | 0,0138 | | 2,09E-08 | 3,00E-08 |
| 470000 | | 0,0097 | 0,0144 | | 2,06E-08 | 3,06E-08 |
| 480000 | | 0,0092 | 0,015 | | 1,92E-08 | 3,13E-08 |
| 490000 | | 0,0095 | 0,0143 | | 1,94E-08 | 2,92E-08 |
| 500000 | | 0,0099 | 0,0143 | | 1,98E-08 | 2,86E-08 |
| 510000 | | 0,0102 | 0,0142 | | 2,00E-08 | 2,78E-08 |
| 520000 | | 0,0098 | 0,0143 | | 1,88E-08 | 2,75E-08 |
| 530000 | | 0,0104 | 0,0138 | | 1,96E-08 | 2,60E-08 |
| 540000 | | 0,01 | 0,0141 | | 1,85E-08 | 2,61E-08 |
| 550000 | | 0,0098 | 0,0137 | | 1,78E-08 | 2,49E-08 |
| 560000 | | 0,0095 | 0,014 | | 1,70E-08 | 2,50E-08 |
| 570000 | | 0,0101 | 0,0145 | | 1,77E-08 | 2,54E-08 |
| 580000 | | 0,0099 | 0,0145 | | 1,71E-08 | 2,50E-08 |
| 590000 | | 0,0103 | 0,0148 | | 1,75E-08 | 2,51E-08 |
| 600000 | | 0,0099 | 0,0148 | | 1,65E-08 | 2,47E-08 |
| 610000 | | 0,0101 | 0,015 | | 1,66E-08 | 2,46E-08 |
| 620000 | | 0,0092 | 0,0155 | | 1,48E-08 | 2,50E-08 |
| 630000 | | 0,0099 | 0,0155 | | 1,57E-08 | 2,46E-08 |
| 640000 | | 0,0097 | 0,0152 | | 1,52E-08 | 2,38E-08 |
| 650000 | | 0,0096 | 0,0148 | | 1,48E-08 | 2,28E-08 |
| 660000 | | 0,0097 | 0,0151 | | 1,47E-08 | 2,29E-08 |
| 670000 | | 0,0098 | 0,0155 | | 1,46E-08 | 2,31E-08 |
| 680000 | | 0,0098 | 0,0146 | | 1,44E-08 | 2,15E-08 |
| 690000 | | 0,0096 | 0,0147 | | 1,39E-08 | 2,13E-08 |
| 700000 | | 0,0098 | 0,0148 | | 1,40E-08 | 2,11E-08 |
| 710000 | | 0,0099 | 0,0146 | | 1,39E-08 | 2,06E-08 |
| 720000 | | 0,0103 | 0,0147 | | 1,43E-08 | 2,04E-08 |
| 730000 | | 0,0107 | 0,0158 | | 1,47E-08 | 2,16E-08 |
| 740000 | | 0,0112 | 0,0157 | | 1,51E-08 | 2,12E-08 |
| 750000 | | 0,012 | 0,0161 | | 1,60E-08 | 2,15E-08 |
| 760000 | | 0,0113 | 0,0152 | | 1,49E-08 | 2,00E-08 |
| 770000 | | 0,0113 | 0,0148 | | 1,47E-08 | 1,92E-08 |
| 780000 | | 0,01 | 0,0148 | | 1,28E-08 | 1,90E-08 |
| 790000 | | 0,0104 | 0,0149 | | 1,32E-08 | 1,89E-08 |
| 800000 | | 0,0099 | 0,0155 | | 1,24E-08 | 1,94E-08 |
| 810000 | | 0,0097 | 0,0149 | | 1,20E-08 | 1,84E-08 |
| 820000 | | 0,0101 | 0,017 | | 1,23E-08 | 2,07E-08 |
| 830000 | | 0,0105 | 0,0177 | | 1,27E-08 | 2,13E-08 |
| 840000 | | 0,0106 | 0,016 | | 1,26E-08 | 1,90E-08 |
| 850000 | | 0,0105 | 0,016 | | 1,24E-08 | 1,88E-08 |
| 860000 | | 0,0103 | 0,0153 | | 1,20E-08 | 1,78E-08 |
| 870000 | | 0,0109 | 0,0161 | | 1,25E-08 | 1,85E-08 |
| 880000 | | 0,0109 | 0,0166 | | 1,24E-08 | 1,89E-08 |
| 890000 | | 0,0106 | 0,016 | | 1,19E-08 | 1,80E-08 |
| 900000 | | 0,011 | 0,0159 | | 1,22E-08 | 1,77E-08 |
| 910000 | | 0,0104 | 0,0159 | | 1,14E-08 | 1,75E-08 |
| 920000 | | 0,0106 | 0,0157 | | 1,15E-08 | 1,71E-08 |
| 930000 | | 0,0104 | 0,0157 | | 1,12E-08 | 1,69E-08 |
| 940000 | | 0,0111 | 0,0159 | | 1,18E-08 | 1,69E-08 |
| 950000 | | 0,0116 | 0,0163 | | 1,22E-08 | 1,72E-08 |
| 960000 | | 0,0111 | 0,0167 | | 1,16E-08 | 1,74E-08 |
| 970000 | | 0,0103 | 0,017 | | 1,06E-08 | 1,75E-08 |
| 980000 | | 0,0108 | 0,0162 | | 1,10E-08 | 1,65E-08 |
| 990000 | | 0,0114 | 0,0162 | | 1,15E-08 | 1,64E-08 |

4. Times related to measuring worst case scenario for BST:

| WORST CASE | Adding | | | Searching | | | Deleting | | |
|---|---|---|---|---|---|---|---|---|---|
| iterator | AVL | LL | BST | AVL | LL | BST | AVL | LL | BST |
| 1000 | 0,0008 | 0,0009 | 0,006 | 0,0002 | 0,0026 | 0,0028 | 0,0008 | 0,002 | 0,276 |
| 2000 | 0,0008 | 0,0044 | 0,0278 | 0,0002 | 0,0062 | 0,0142 | 0,0002 | 0,0056 | 0,2898 |
| 3000 | 0,0006 | 0,0094 | 0,0282 | 0,0012 | 0,0103 | 0,0278 | 0,0006 | 0,01 | 0,308 |
| 4000 | 0,0006 | 0,0104 | 0,0404 | 0,0012 | 0,0132 | 0,0532 | 0,0006 | 0,0159 | 0,3064 |
| 5000 | 0,001 | 0,0144 | 0,0528 | 0,0008 | 0,0177 | 0,0878 | 0,0004 | 0,0227 | 0,319 |
| 6000 | 0,0006 | 0,0201 | 0,0628 | 0,0006 | 0,0244 | 0,1286 | 0,0008 | 0,03 | 0,326 |
| 7000 | 0,0008 | 0,0261 | 0,0738 | 0,0012 | 0,0308 | 0,1778 | 0,001 | 0,038 | 0,333 |
| 8000 | 0,0014 | 0,0348 | 0,0868 | 0,002 | 0,0402 | 0,2384 | 0,001 | 0,0458 | 0,3458 |
| 9000 | 0,0006 | 0,0417 | 0,0976 | 0,0012 | 0,0492 | 0,3018 | 0,0006 | 0,0541 | 0,3512 |
| 10000 | 0,0008 | 0,0485 | 0,1082 | 0,0016 | 0,0566 | 0,3772 | 0,0006 | 0,0608 | 0,3588 |
| 11000 | 0,0004 | 0,056 | 0,1192 | 0,0018 | 0,0648 | 0,4572 | 0,0008 | 0,0677 | 0,3588 |
| 12000 | 0,0008 | 0,0633 | 0,1298 | 0,0022 | 0,0713 | 0,5472 | 0,0006 | 0,0755 | 0,3728 |
| 13000 | 0,0012 | 0,0719 | 0,1414 | 0,003 | 0,0802 | 0,6424 | 0,0006 | 0,083 | 0,385 |
| 14000 | 0,001 | 0,0768 | 0,156 | 0,0036 | 0,0866 | 0,751 | 0,0006 | 0,09 | 0,377 |
| 15000 | 0,001 | 0,0858 | 0,1654 | 0,0028 | 0,0938 | 0,8608 | 0,001 | 0,0977 | 0,3854 |
| 16000 | 0,0008 | 0,0901 | 0,1762 | 0,003 | 0,1 | 0,9826 | 0,0004 | 0,1054 | 0,4018 |
| 17000 | 0,0014 | 0,0973 | 0,19 | 0,004 | 0,1099 | 1,1252 | 0,0008 | 0,1131 | 0,4082 |
| 18000 | 0,0008 | 0,1049 | 0,2004 | 0,0042 | 0,1165 | 1,2482 | 0,001 | 0,1226 | 0,4318 |
| 19000 | 0,0004 | 0,1118 | 0,2124 | 0,0044 | 0,1241 | 1,3998 | 0,002 | 0,1313 | 0,438 |
| 20000 | 0,001 | 0,1186 | 0,227 | 0,0042 | 0,1311 | 1,5568 | 0,0012 | 0,1373 | 0,4378 |