

# Analysis Explanation.

## BST

### For Large and Small File

#### Sorted.txt

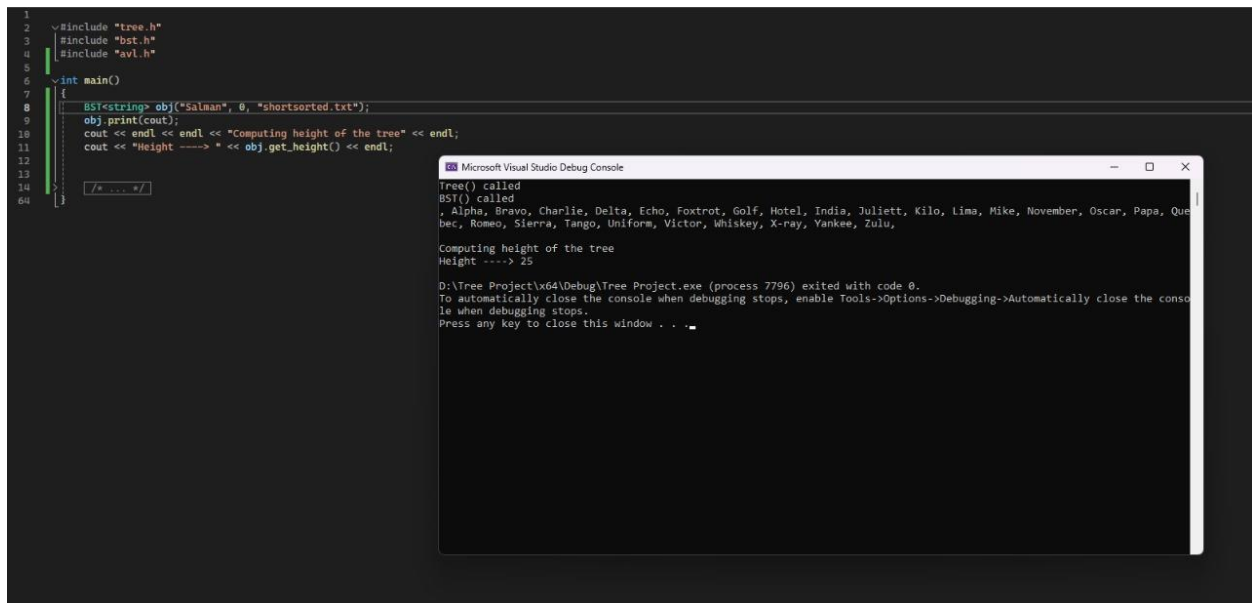
When I try to run the code on sorted.txt using a BST, it goes two ways. Either it throws an exception on stack overflow or it generates a tree which is either left skewed or right skewed with a height of 45000+.

This is the worst case of this data structure as it is now behaving as a Linked List instead of a Tree.

Its searching operation will now take  $O(N)$  instead of  $\log(N)$ .

## Examples:

### Shortsorted.txt

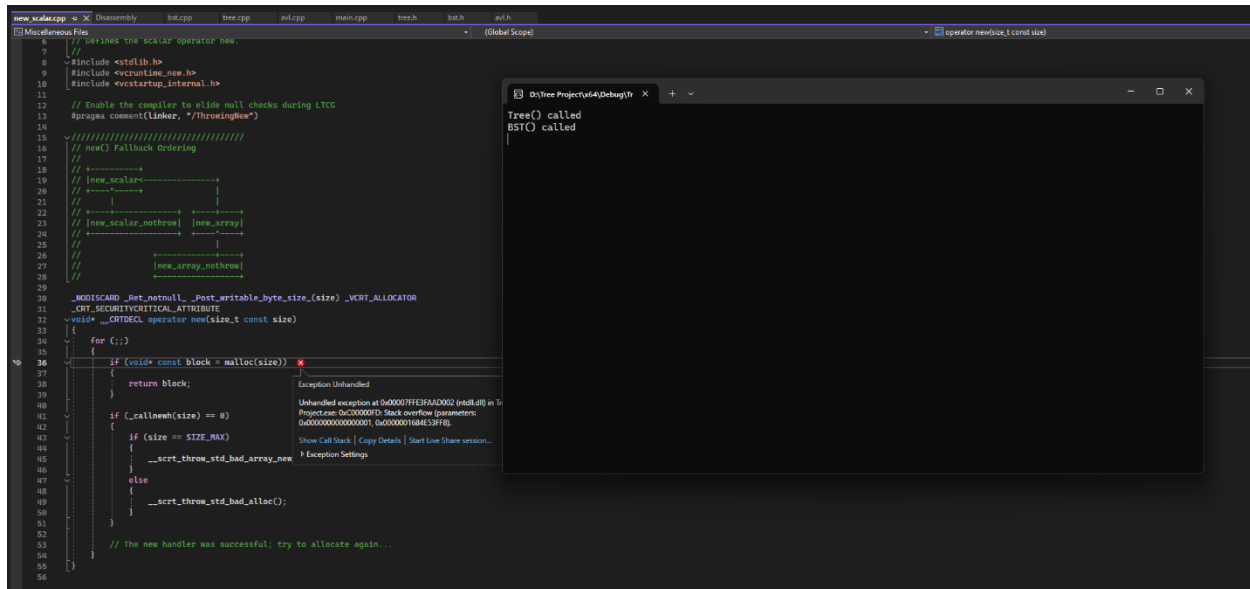


```
1
2  #include "tree.h"
3  #include "bst.h"
4  #include "avl.h"
5
6  int main()
7  {
8      BST<string> obj("Salman", 0, "shortsorted.txt");
9      obj.print(cout);
10     cout << endl << endl << "Computing height of the tree" << endl;
11     cout << "Height ----> " << obj.get_height() << endl;
12
13
14     /s...s/
15
64 }
```

```
Microsoft Visual Studio Debug Console
Tree() called
BST() called
. Alpha, Bravo, Charlie, Delta, Echo, Foxtrot, Golf, Hotel, India, Juliett, Kilo, Lima, Mike, November, Oscar, Papa, Quebec, Romeo, Sierra, Tango, Uniform, Victor, Whiskey, X-ray, Yankee, Zulu,
Computing height of the tree
Height ----> 25
D:\Tree Project\Debug\Tree Project.exe (process 7796) exited with code 0.
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the console when debugging stops.
Press any key to close this window . . .
```

As you can clearly see that height of our tree is, 25 for a total count of 25 Nodes. It indicates that tree is skewed to only one side and will take  $O(N)$  instead of  $\log(n)$ .

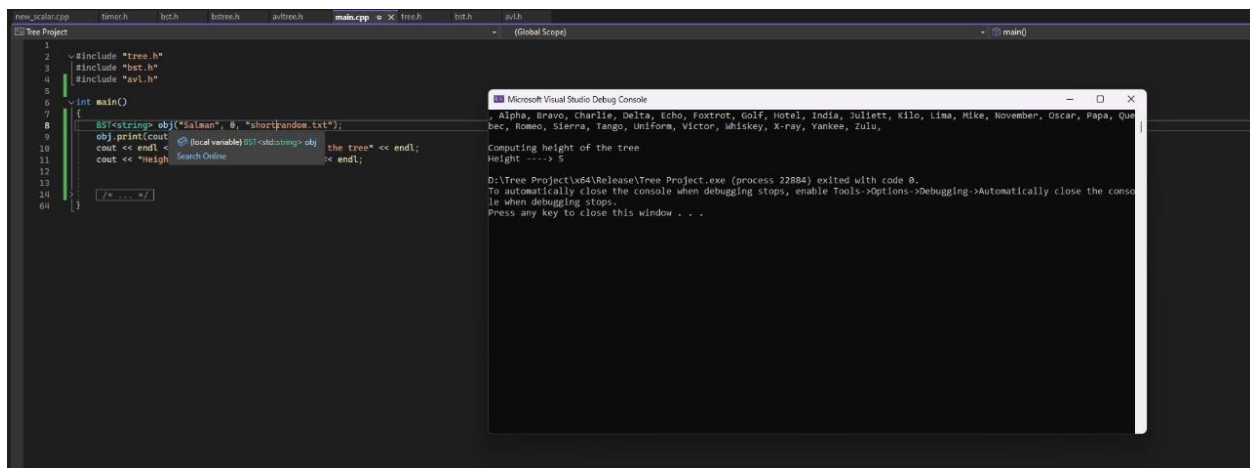
Sorted.txt



For this dataset, compiler throws an StackOverflow error, because tree is just growing in one direction and behaving like a linked list. For this case our height would be 45393 and Count will also be 45393.

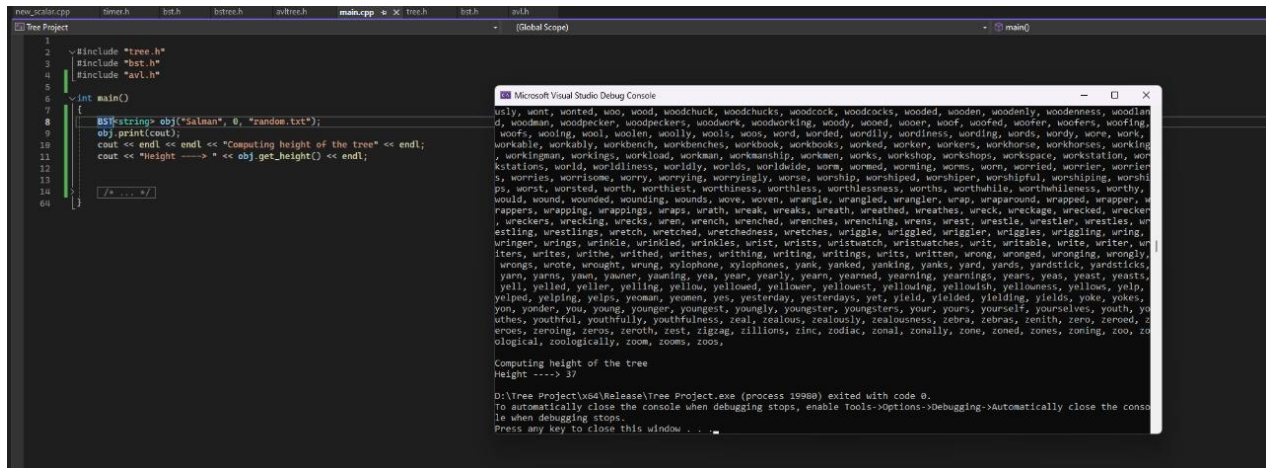
This indicates the **limitations of a BST Tree when data is sorted.**

ShortRandom.txt



For this purpose, BST has a height of 5 with a total node count of 25. As data is random, tree grows on both sides and achieve a height of 5 on which search operation would take place as  $\log(n)$ .

Random.txt



The screenshot shows a C++ program in Visual Studio. The code in `main.cpp` includes `<tree.h>`, `<bst.h>`, and `<avl.h>`. It defines a `main` function that creates a `BST` object named `obj` with the filename "random.txt" and a seed of 0. It then prints the tree and its height. The output in the console shows the tree structure and a height of 37. A list of words is also displayed in the console window.

```
1 #include "tree.h"
2 #include "bst.h"
3 #include "avl.h"
4
5 int main()
6 {
7     BST<string> obj("salman", 0, "random.txt");
8     obj.print(out);
9     cout << endl << endl << "Computing height of the tree" << endl;
10    cout << "Height ----> " << obj.get_height() << endl;
11
12    // ...
13
14    return 0;
15 }
```

Microsoft Visual Studio Debug Console

uily, wont, wonted, woo, wood, woodchuck, woodchucks, woodcock, woodcocks, wooded, wooden, woodenly, woodenness, woodles  
d, woodman, woodpecker, woodpeckers, woodwork, woodworking, woody, wooed, wooser, woof, woofed, woofers, woofing, woofs, woofing, woo, wooler, woolly, wool, woot, word, worded, wordily, wordiness, wording, words, wordy, wore, work, workable, workably, workbench, workbenches, workbook, workbooks, worked, worker, workers, workhorse, workhorses, working, workingman, workings, workload, workman, workmanship, workmen, works, workshop, workshops, workspace, workstation, work  
itations, world, worldliness, worldly, worlds, worldwide, worn, wormed, worming, worms, worn, worried, worrier, worrier  
s, worries, worrisome, worry, worrying, worryingly, worra, worship, worshiped, worshiper, worshipful, worshipping, worshi  
ps, worst, worsted, worth, worthiest, worthiness, worthless, worthlessness, worths, worthwhile, worthwhileness, worthy,  
would, wound, wounded, wounding, wounds, wove, woven, wrangle, wrangled, wrangler, wrap, wraparound, wrapped, wrapper, w  
rappers, wrapping, wrappings, wraps, wrath, wreak, wreaks, wreath, wreathed, wreathes, wreck, wreckage, wrecked, wrecker  
, wreckers, wrecking, wrecks, wran, wrench, wrenched, wrenches, wrenching, wrens, wrest, wrestle, wrestler, wrestles, wr  
estling, wrestlings, wretch, wretched, wretchedness, wretches, wriggle, wriggled, wriggler, wriggles, wriggling, wring,  
wringers, wrings, wrinkle, wrinkled, wrinkles, wrist, wrists, wristwatch, wristwatches, writ, writable, write, writer, w  
riters, writes, writhe, writed, writhes, writhing, writing, writings, writs, written, wrong, wronged, wrongly, wrongs, wrote, wrought, wrung, xylophone, xylophones, yank, yanked, yanking, yanks, yard, yards, yardstick, yardsticks,  
yarn, yarns, yarn, yarner, yarning, yea, year, yearly, yearn, yearned, yearning, yearnings, years, yeas, yeast, yeasts, yell, yelled, yellor, yelling, yellow, yellowed, yellower, yellowest, yellowing, yellowish, yellowness, yellow, yelp, yelped, yelping, yelps, yeoman, yeoman, yes, yesterday, yesterdays, yet, yield, yielded, yielding, yields, yoke, yokes, you, yonder, you, young, younger, youngest, youngl, youngster, youngsters, your, yours, yourself, yourselves, youth, y  
uthes, youthful, youthfully, youthfulness, zeal, zealous, zealously, zealousness, zebra, zebras, zenith, zero, zeroes, zeroes, zeroing, zeros, zeroth, zest, zigzag, zillions, zinc, zodiac, zonal, zonally, zone, zoned, zones, zoning, zoo, zo  
ological, zoologically, zoom, zooms, zoos.

Computing height of the tree  
Height ----> 37

D:\Tree Project\Release\Tree Project.exe (process 19980) exited with code 0.  
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the console when debugging stops.  
Press any key to close this window . . .

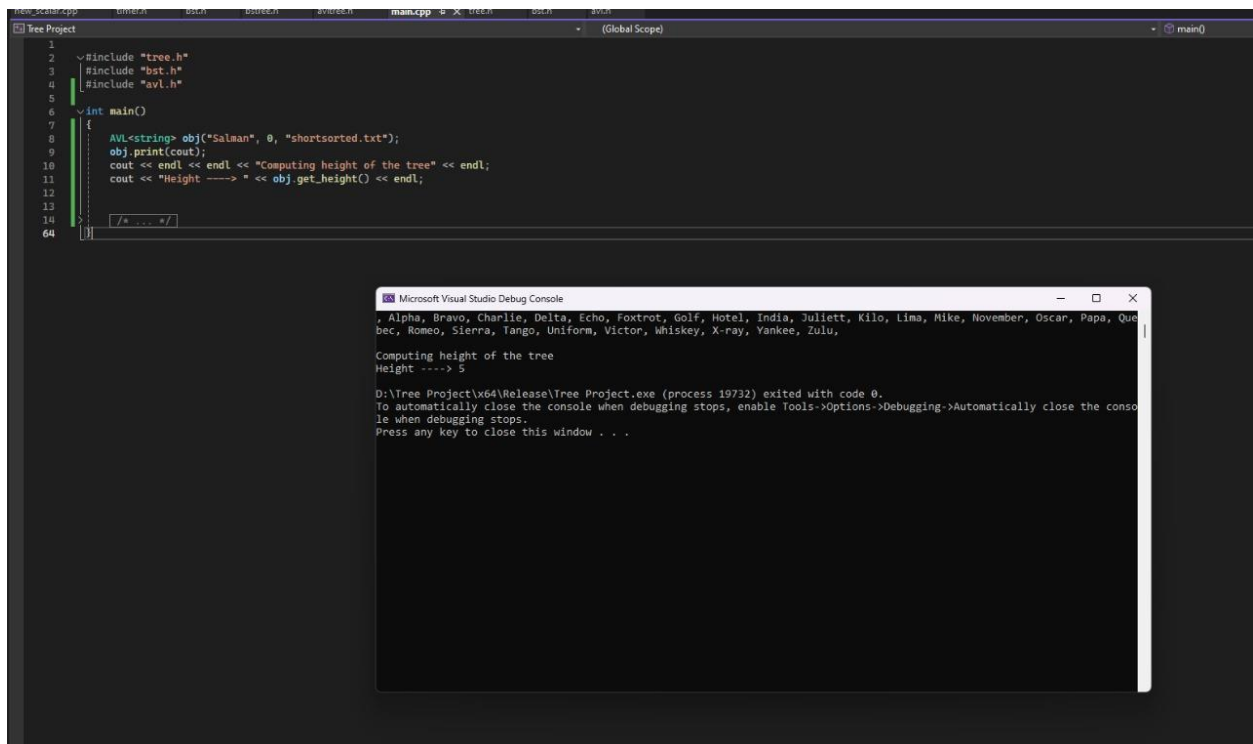
For this purpose, as data is random BST has a height of 37 with a total node count of 45393. This shows the power of BST when data is random. For this node count if we want to search anything, it would happen in  $\log(n)$ .

# AVL

For Large and Small Files.

Examples.

*ShortSorted.txt*



```
1 2
2 3
3 4
4 5
5 6
6 7
7 8
8 9
9 10
10 11
11 12
12 13
13 14
14 64
```

```
#include "tree.h"
#include "bst.h"
#include "avl.h"

int main()
{
    AVL<string> obj("Salman", 0, "shortsorted.txt");
    obj.print(cout);
    cout << endl << endl << "Computing height of the tree" << endl;
    cout << "Height ----> " << obj.get_height() << endl;
}
```

```
Microsoft Visual Studio Debug Console
, Alpha, Bravo, Charlie, Delta, Echo, Foxtrot, Golf, Hotel, India, Juliett, Kilo, Lima, Mike, November, Oscar, Papa, Quebec, Romeo, Sierra, Tango, Uniform, Victor, Whiskey, X-ray, Yankee, Zulu,

Computing height of the tree
Height ----> 5

D:\Tree Project\x64\Release\Tree Project.exe (process 19732) exited with code 0.
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the console when debugging stops.
Press any key to close this window . . .
```

As you can clearly see that height of our tree is, 5 for a total count of 25 Nodes. It indicates that tree is self-balancing itself and will perform  $\text{Log}(N)$  on search operations.

*Sorted.txt*

```
1 ~#include "tree.h"
2 ~#include "bst.h"
3 ~#include "avl.h"
4
5 ~int main()
6 {
7     AVL<string> obj("Salman", 0, "sorted.txt");
8     obj.print(cout);
9     cout << endl << endl << "Computing height of the tree" << endl;
10    cout << "Height ----> " << obj.get_height() << endl;
11
12    // ...
13
14 }
```

```
usly, wont, wonted, woo, wood, woodchuck, woodchucks, woodcock, woodcocks, wooded, wooden, woodenly, woodenness, woodlan
d, woodman, woodpecker, woodpeckers, woodwork, woodworking, woody, wooed, woof, woofed, woofers, woofing,
woof, woofing, wool, woollen, woolly, woofs, woos, word, worded, wordily, wordiness, wording, words, wordy, wore, work,
workable, workably, workbench, workbenches, workbook, workbooks, worked, worker, workers, workhorse, workhorses, working
, workingman, workings, workload, workman, workmanship, workmen, works, workshop, workshops, workspace, workstation, wor
kstations, world, worldliness, worldly, worldie, worldwide, worn, worned, wearing, wears, worn, worried, worrier, worrier
s, worries, worrisome, worry, worrying, worryingly, worse, worship, worshiped, worshiper, worshipful, worshipping, worshi
ps, worst, worsted, worth, worthiest, worthiness, worthless, worthlessness, worths, worthwhile, worthwhileness, worthy,
would, wound, wounded, wounding, wounds, wove, woven, wrangle, wrangled, wrangler, wrap, wraparound, wrapped, wrapper, w
rappers, wrapping, wrappings, wrap, wrath, wreak, wreaks, wreak, wreaked, wreaks, wreck, wreckage, wrecked, wrecker,
wreckers, wrecking, wrecks, wren, wrench, wrenched, wrenches, wrenching, wrens, wrest, wrestle, wrestler, wrestles, wr
estling, wrestlings, wretch, wretched, wretchedness, wretches, wriggle, wriggled, wriggler, wriggles, wriggling, wring,
winger, wrings, wrinkle, wrinkled, wrinkles, wrist, wrists, wristbitch, wristbatches, writ, writable, write, writer, wri
ters, writes, writhe, writhed, writhes, writhing, writing, writings, writs, written, wrong, wronged, wronging, wrongly,
wrongs, wrote, wrought, wrung, xylophone, xylophones, yank, yanked, yanking, yanks, yard, yards, yardstick, yardsticks,
yarn, yarns, yarn, yarner, yawning, yes, year, yearly, yearn, yearned, yearning, yearnings, years, year, yeast, yeasts,
yell, yelled, yellor, yelling, yellow, yellowed, yellower, yellowest, yellowing, yellowish, yellowness, yellows, yelp,
yelped, yelping, yelps, yeoman, yeomen, yes, yesterday, yesterdays, yet, yield, yielded, yielding, yields, yoke, yokes,
yon, yonder, you, young, younger, youngest, youngly, youngster, youngsters, your, yours, yourself, yourselves, youth, yo
uths, youthful, youthfully, youthfulness, zeal, zealous, zealously, zealousness, zebra, zebras, zeith, zero, zeroed, z
eros, zeroing, zeros, zeroth, zest, zigzag, zillions, zinc, zodiac, zonal, zonally, zone, zoned, zones, zoning, zoo, zo
ological, zoologically, zoom, zooms, zoos.
```

```
Computing height of the tree
Height ----> 15

D:\Free Project\src\Release\Free Project.exe (process 19288) exited with code 0.
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the console
when debugging stops.
Press any key to close this window . . .
```

For this dataset, AVL expands to a height of 15 for a total of 45393 nodes. This result is remarkably good and will give us any Node in  $\log(N)$ .

This indicates the **Power of a AVL Tree when data is sorted.**

*ShortRandom.txt*

```
1 ~#include "tree.h"
2 ~#include "bst.h"
3 ~#include "avl.h"
4
5 ~int main()
6 {
7     AVL<string> obj("Salman", 0, "shortrandom.txt");
8     obj.print(cout);
9     cout << endl << endl << "Computing height of the tree" << endl;
10    cout << "Height ----> " << obj.get_height() << endl;
11
12    // ...
13
14 }
```

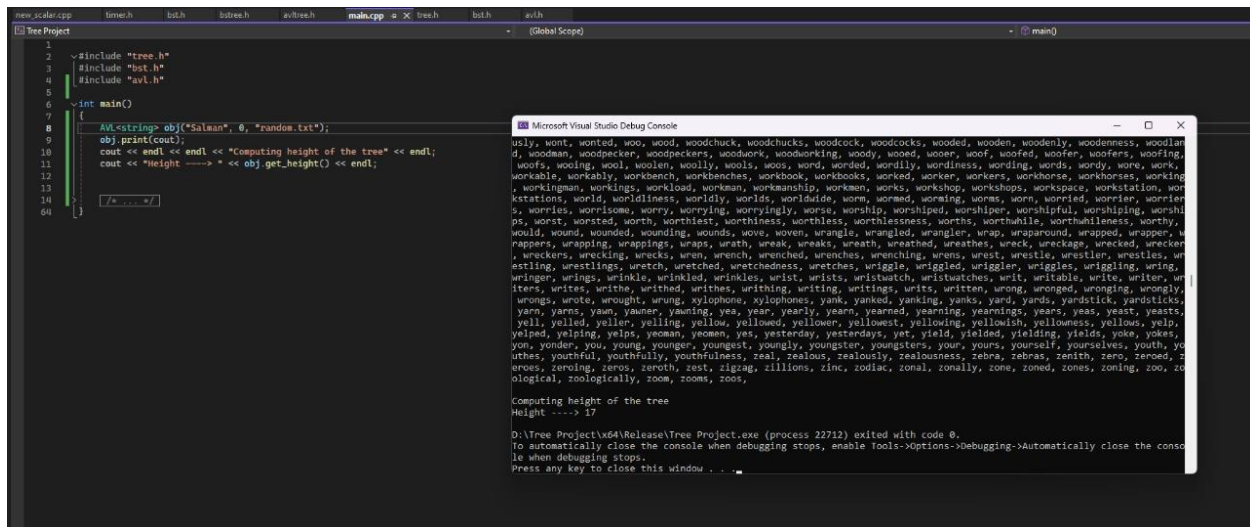
```
, Alpha, Bravo, Charlie, Delta, Echo, Foxtrot, Golf, Hotel, India, Juliett, Kilo, Lima, Mike, November, Oscar, Papa, Que
bec, Romeo, Sierra, Tango, Uniform, Victor, Whiskey, X-ray, Yankee, Zulu,
```

```
Computing height of the tree
Height ----> 4

D:\Free Project\src\Release\Free Project.exe (process 14512) exited with code 0.
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the console
when debugging stops.
Press any key to close this window . . .
```

For this purpose, AVL has a height of 4 with a total node count of 25. As data is random, tree doesn't grow on both sides. Instead, it performs rotations on its own and balance itself.

Random.txt



The screenshot shows a Visual Studio IDE with a C++ project. The main.cpp file contains the following code:

```
1 ~#include "tree.h"
2 ~#include "bst.h"
3 ~#include "avl.h"
4
5 ~int main()
6 {
7     AVL<string> obj("Salman", 0, "random.txt");
8     obj.print(cout);
9     cout << endl << endl << "Computing height of the tree" << endl;
10    cout << "Height ----> " << obj.get_height() << endl;
11
12    // ...
13
14 }
```

The Microsoft Visual Studio Debug Console shows the output of the program:

```
Computing height of the tree
Height ----> 17

D:\Tree Project\64\Release\Tree Project.exe (process 22712) exited with code 0.
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the console when debugging stops.
Press any key to close this window . . .
```

For this purpose, as data is random AVL has a height of 17 with a total node count of 45393. If we compare it to BST which had a height of 37 for the same Dataset, we can conclude that AVL is a far better approach for sorted and random data as it makes sure that our search operations are always in  $\log(N)$ .

THE END.