HW07 Report

By JJ McCauley

METHODS

This assignment aims to compare the differences between AVL trees and Red-Black trees. Each tree was tested the following variables:

- Data Range (Low Bound, High Bound)
- Number of insertion/ deletion cycles

Each tree was tested with 6-10 node sizes, recording the total time, the average IPL before the cycles, and the average IPL after the following samples:

- Data Range = 0-Number of Nodes, Cycles = Number of Nodes
- Data Range = 0-50,000, Cycles = Number of Nodes
- Data Range = 0-Number of Nodes, Cycles = 50,000
- Data Range = 0-50,000, Cycles = 50,000
- Data Range = 0-Number of Nodes, Cycles = 0

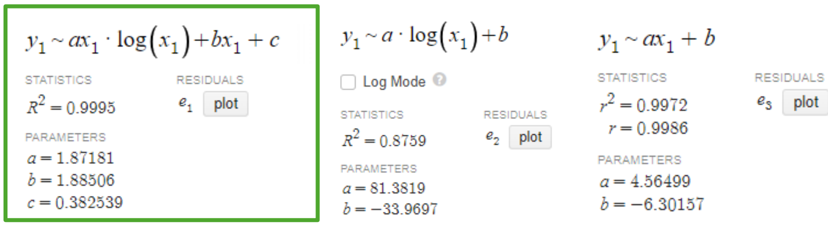
With each sample, a graph of the IPLs and the graph and line of best fit of the timings are included. The best fit line will be highlighted in green. The graph included with the timing includes the line of best fit.

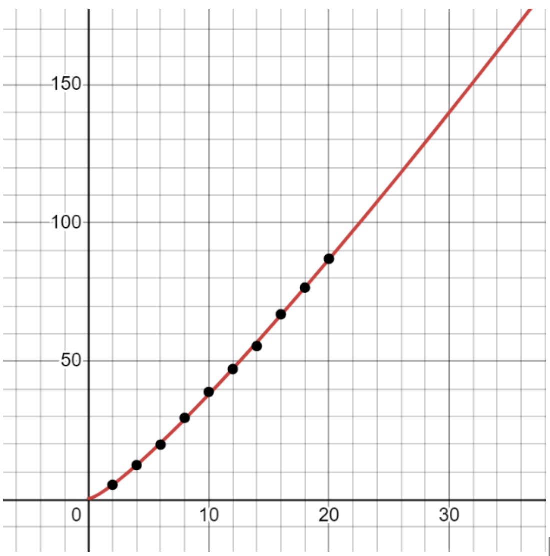
RESULTS

(see next page)

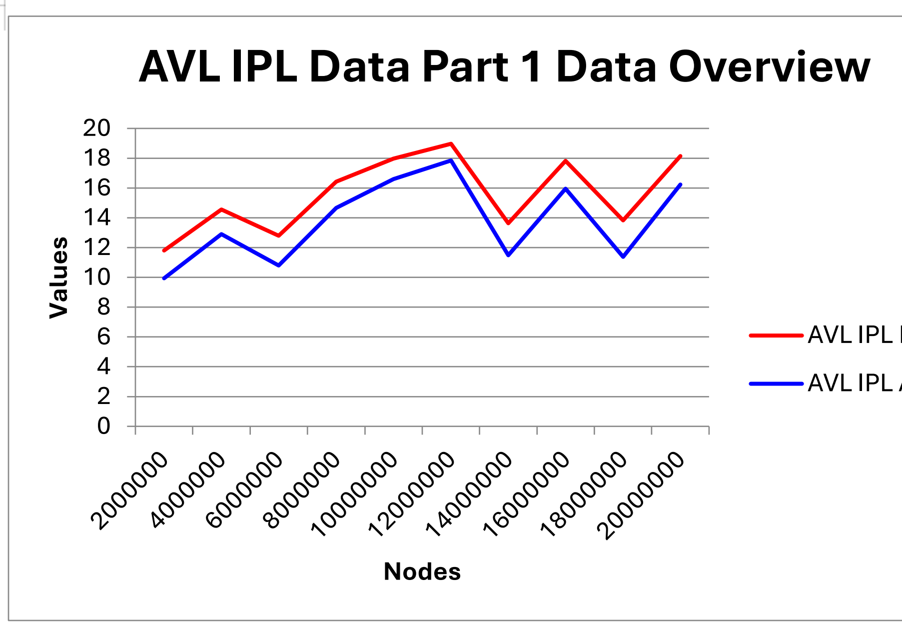
Data Range = 0-Number of Nodes, Cycles = Number of Nodes

AVL Tree

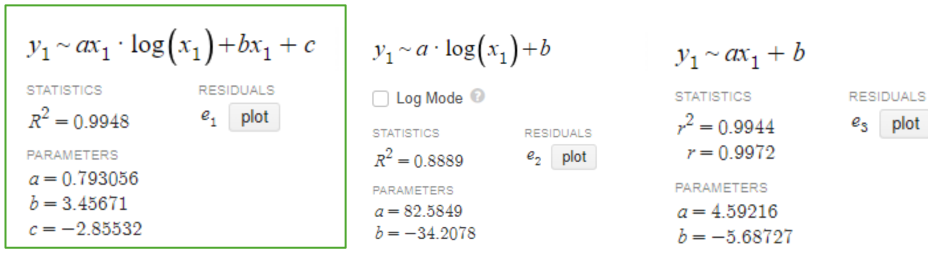


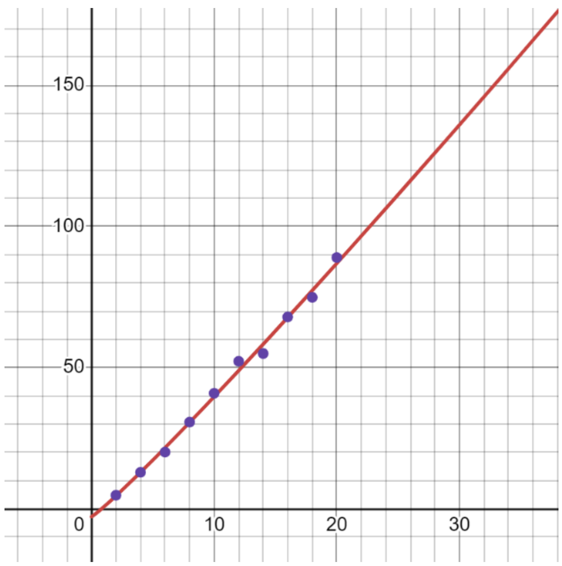


Nodes	AVL IPL Before	AVL IPL After
2000000	11.8045	9.93949
4000000	14.5666	12.8931
6000000	12.8051	10.7912
8000000	16.4242	14.6584
10000000	17.9779	16.6077
12000000	18.9701	17.8484
14000000	13.6278	11.4903
16000000	17.825	15.9614
18000000	13.8371	11.3733
20000000	18.1513	16.2431

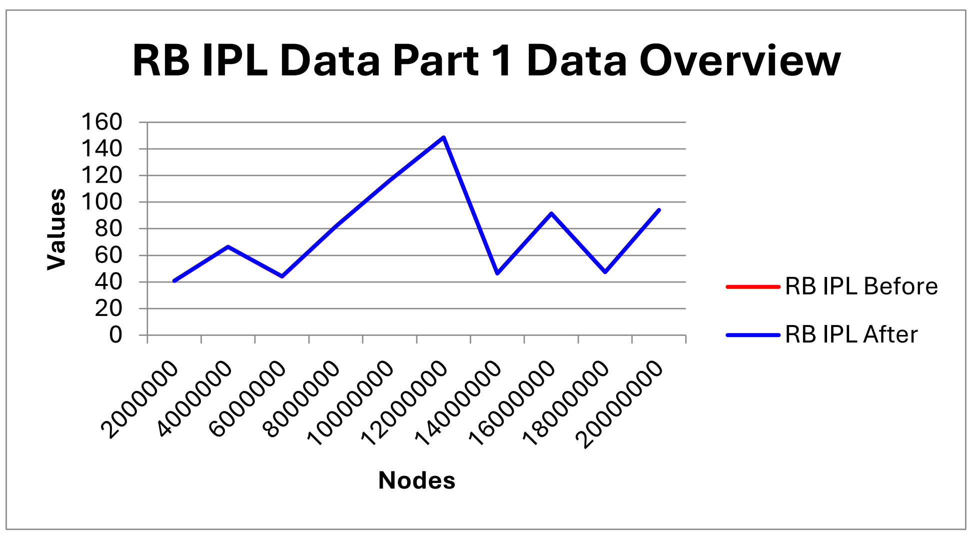


Red-Black Tree





Nodes	RB IPL Before	RB IPL After
2000000	40.8012	40.8644
4000000	66.442	66.2794
6000000	44.1026	44.1615
8000000	81.7249	81.6656
10000000	116.149	116.167
12000000	148.582	148.633
14000000	46.4427	46.5059
16000000	91.433	91.4802
18000000	47.294	47.379
20000000	94.0543	94.1164



Data Range = 0-50,000, Cycles = Number of Nodes

AVL Tree

$y_1 \sim a x_1 \cdot \log(x_1) + b x_1 + c$

STATISTICS

$R^2 = 0.9996$

RESIDUALS

e_1 plot

PARAMETERS

$a = 0.0663102$

$b = 0.689645$

$c = 0.103119$

$y_1 \sim a \cdot \log(x_1) + b$

Log Mode

STATISTICS

$R^2 = 0.9122$

RESIDUALS

e_2 plot

PARAMETERS

$a = 11.9349$

$b = -3.54717$

$y_1 \sim a x_1 + b$

STATISTICS

$r^2 = 0.9995$

$r = 0.9998$

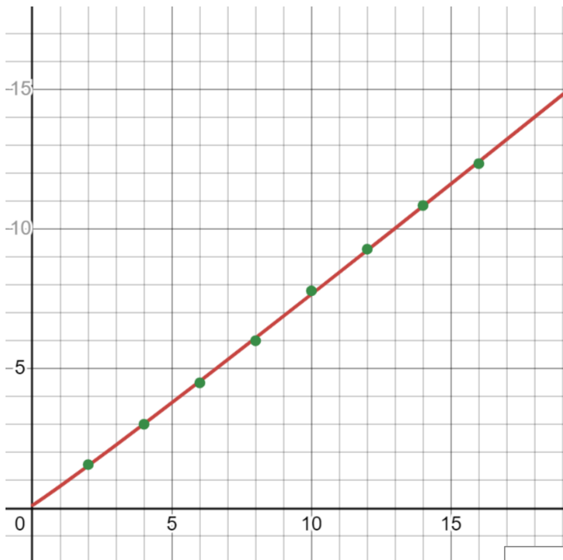
RESIDUALS

e_3 plot

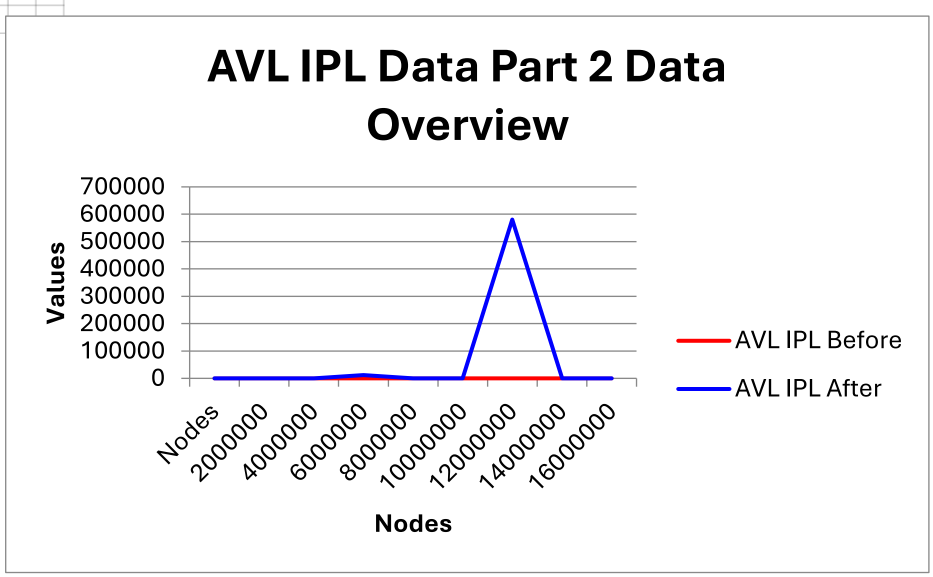
PARAMETERS

$a = 0.779027$

$b = -0.0948654$



Nodes	AVL IPL Before	AVL IPL After
Nodes	AVL IPL Before	AVL IPL After
2000000	0.350313	0.350297
4000000	0.17412	0.17412
6000000	0.11608	11608
8000000	0.08706	0.08706
10000000	0.069658	0.069658
12000000	0.058048	580482
14000000	0.049756	0.049756
16000000	0.043536	0.043536



Red-Black Tree

$y_1 \sim a x_1 \cdot \log(x_1) + b x_1 + c$

STATISTICS

$R^2 = 0.9777$

RESIDUALS

e_1 plot

PARAMETERS

$a = 8.35319$

$b = -5.53168$

$c = 11.2467$

$y_1 \sim a \cdot \log(x_1) + b$

Log Mode

STATISTICS

$R^2 = 0.8005$

RESIDUALS

e_2 plot

PARAMETERS

$a = 84.2184$

$b = -35.9784$

$y_1 \sim a x_1 + b$

STATISTICS

$r^2 = 0.9523$

$r = 0.9759$

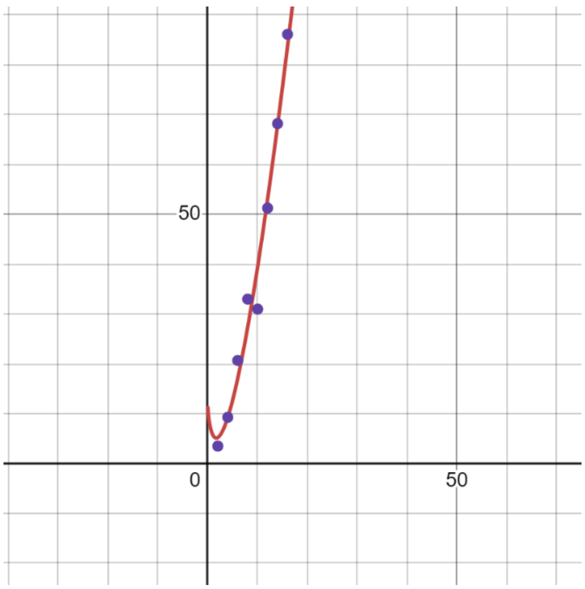
RESIDUALS

e_3 plot

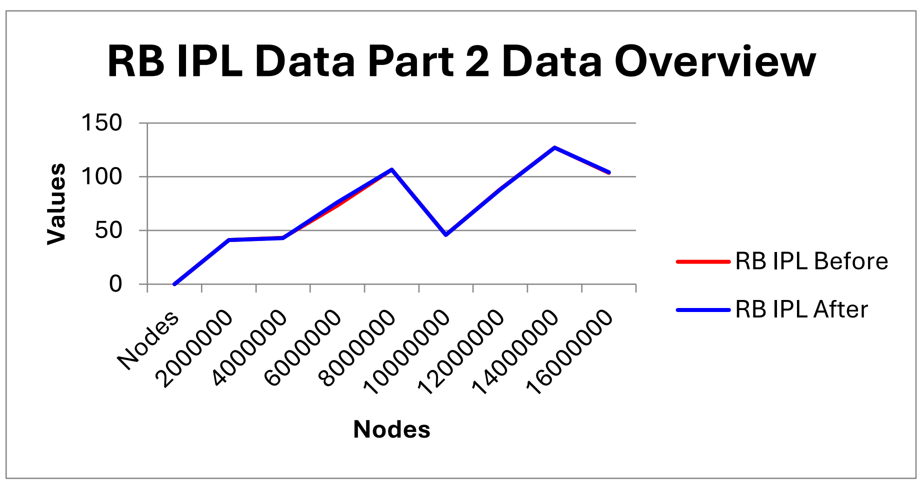
PARAMETERS

$a = 5.72792$

$b = -13.6937$



Nodes	RB IPL Before	RB IPL After
Nodes	RB IPL Before	RB IPL After
2000000	41.1451	41.0022
4000000	43.1241	42.9929
6000000	73.3409	76.164
8000000	106.689	106.635
10000000	45.8843	45.7128
12000000	88.3331	88.1201
14000000	127.162	127.107
16000000	103.784	104.294



Data Range = 0-Number of Nodes, Cycles = 50,000

AVL Tree

$y_1 \sim ax_1 \cdot \log(x_1) + bx_1 + c$

STATISTICS

$R^2 = 0.9998$

RESIDUALS

e_1 plot

PARAMETERS

$a = 0.660154$
$b = 0.602175$
$c = -0.455627$

$y_1 \sim a \cdot \log(x_1) + b$

Log Mode

STATISTICS

$R^2 = 0.8896$

RESIDUALS

e_2 plot

PARAMETERS

$a = 22.598$
$b = -8.8106$

$y_1 \sim ax_1 + b$

STATISTICS

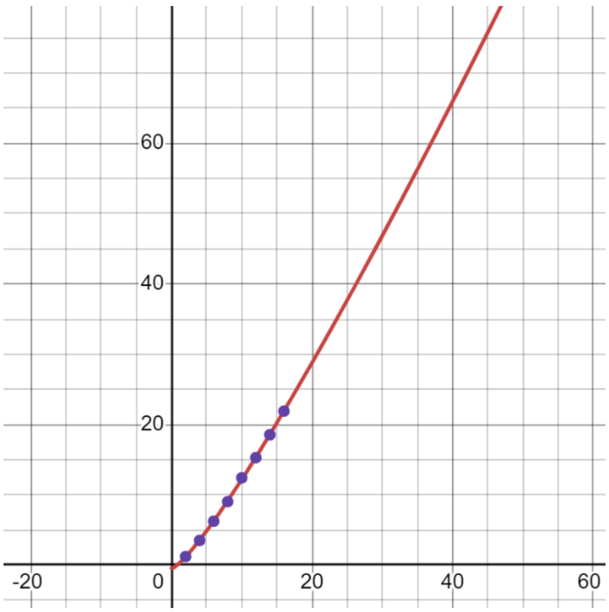
$r^2 = 0.9973$
$r = 0.9987$

RESIDUALS

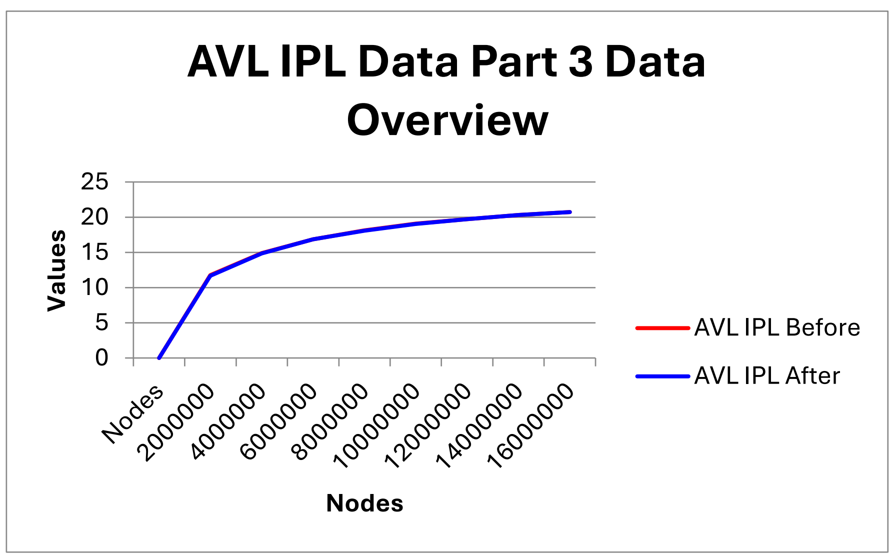
e_3 plot

PARAMETERS

$a = 1.49202$
$b = -2.42667$



Nodes	AVL IPL Before	AVL IPL After
Nodes	AVL IPL Before	AVL IPL After
2000000	11.7813	11.6782
4000000	14.9039	14.8576
6000000	16.8604	16.8347
8000000	18.1052	18.0894
10000000	19.0645	19.055
12000000	19.7394	19.7324
14000000	20.2941	20.2884
16000000	20.7249	20.7211



Red-Black Tree

$y_1 \sim a x_1 \cdot \log(x_1) + b x_1 + c$

STATISTICS

$R^2 = 0.9998$

RESIDUALS

e_1 plot

PARAMETERS

$a = 2.09285$

$b = -0.801897$

$c = 1.49291$

$y_1 \sim a \cdot \log(x_1) + b$

Log Mode

STATISTICS

$R^2 = 0.8472$

RESIDUALS

e_2 plot

PARAMETERS

$a = 30.0064$

$b = -12.8907$

$y_1 \sim a x_1 + b$

STATISTICS

$r^2 = 0.9865$

$r = 0.9933$

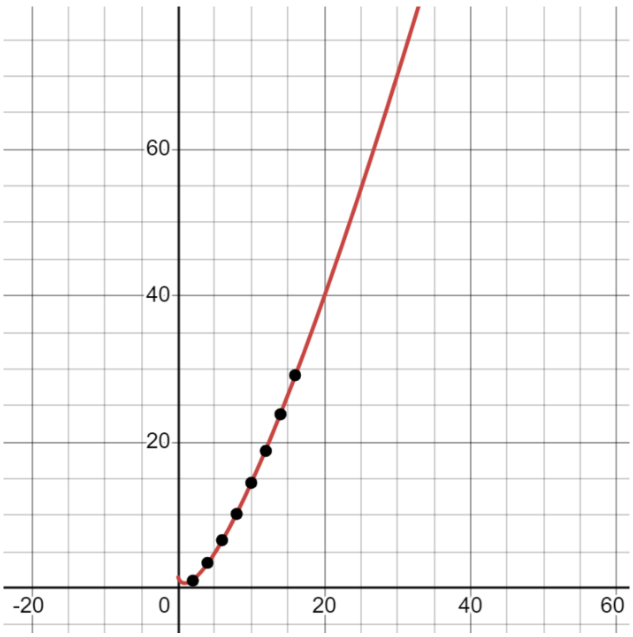
RESIDUALS

e_3 plot

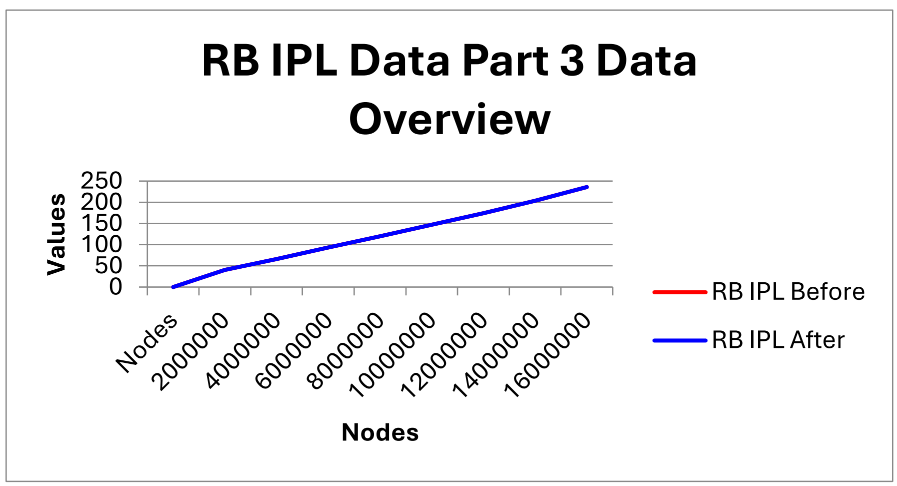
PARAMETERS

$a = 2.01914$

$b = -4.75576$



Nodes	RB IPL Before	RB IPL After
Nodes	RB IPL Before	RB IPL After
2000000	40.8159	40.818
4000000	66.1849	66.1885
6000000	92.9915	92.9932
8000000	119.864	119.865
10000000	146.614	146.615
12000000	174.324	174.323
14000000	203.709	203.709
16000000	235.9159	235.916



Data Range = 0-50,000, Cycles = 50,000

AVL Tree

$y_1 \sim ax_1 \cdot \log(x_1) + bx_1 + c$

STATISTICS

$R^2 = 0.8775$

RESIDUALS

e_1 plot

PARAMETERS

$a = -0.184279$

$b = 0.512296$

$c = -0.154186$

$y_1 \sim a \cdot \log(x_1) + b$

Log Mode

STATISTICS

$R^2 = 0.8735$

RESIDUALS

e_2 plot

PARAMETERS

$a = 6.82957$

$b = -3.40492$

$y_1 \sim ax_1 + b$

STATISTICS

$r^2 = 0.8704$

$r = 0.933$

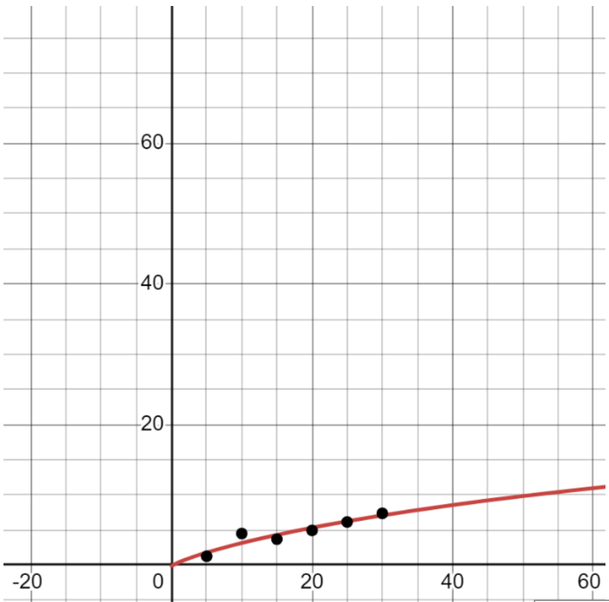
RESIDUALS

e_3 plot

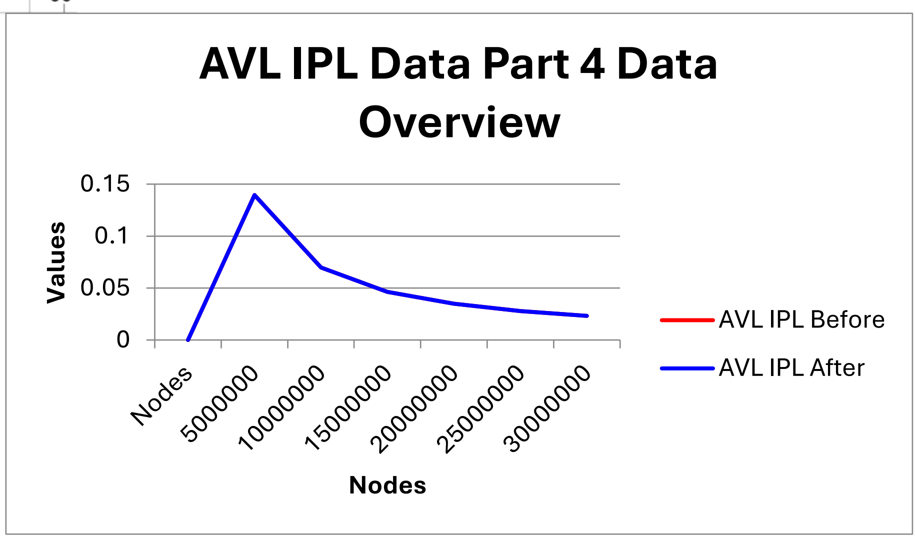
PARAMETERS

$a = 0.209743$

$b = 0.950635$



Nodes	AVL IPL Before	AVL IPL After
Nodes	AVL IPL Before	AVL IPL After
5000000	0.139477	0.139477
10000000	0.069565	0.069597
15000000	0.046398	0.046398
20000000	0.034798	0.034798
25000000	0.027839	0.027839
30000000	0.023199	0.023199



Red-Black Tree

$y_1 \sim ax_1 \cdot \log(x_1) + bx_1 + c$

STATISTICS

$R^2 = 0.9997$

RESIDUALS

e_1 plot

PARAMETERS

$a = 3.68775$
$b = -3.50453$
$c = 7.93708$

$y_1 \sim a \cdot \log(x_1) + b$

Log Mode

STATISTICS

$R^2 = 0.846$

RESIDUALS

e_2 plot

PARAMETERS

$a = 77.0874$
$b = -60.1383$

$y_1 \sim ax_1 + b$

STATISTICS

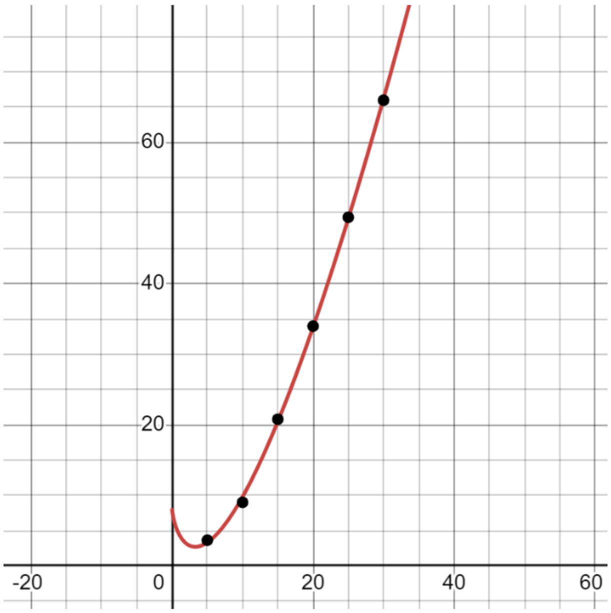
$r^2 = 0.9782$
$r = 0.989$

RESIDUALS

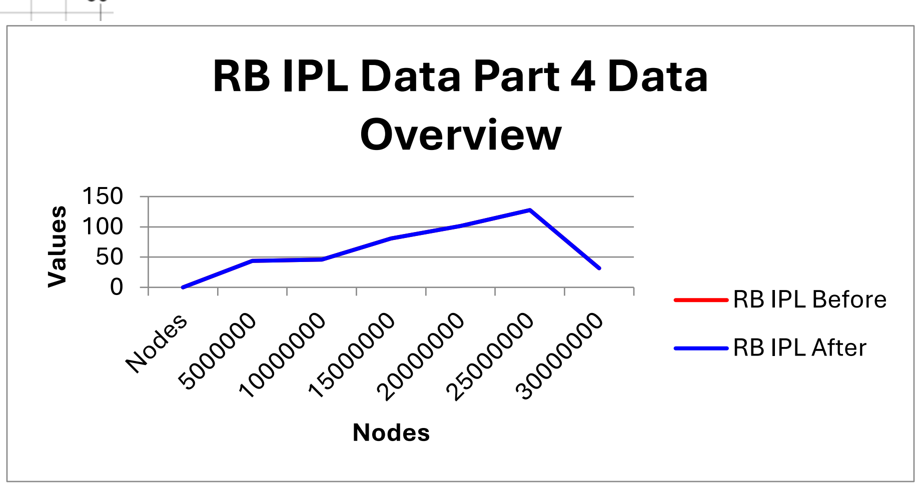
e_3 plot

PARAMETERS

$a = 2.55009$
$b = -14.1723$



Nodes	RB IPL Before	RB IPL After
Nodes	RB IPL Before	RB IPL After
5000000	43.8453	43.6547
10000000	45.8928	45.8928
15000000	80.8255	80.6558
20000000	101.51	101.599
25000000	127.6109	127.6422
30000000	31.811	31.7879



Data Range = 0-Number of Nodes, Cycles = 0

AVL Tree

$y_1 \sim ax_1 \cdot \log(x_1) + bx_1 + c$

STATISTICS

$R^2 = 0.9983$

RESIDUALS

e_1 plot

PARAMETERS

$a = 0.701428$

$b = 0.496602$

$c = -1.17633$

$y_1 \sim a \cdot \log(x_1) + b$

Log Mode

STATISTICS

$R^2 = 0.8911$

RESIDUALS

e_2 plot

PARAMETERS

$a = 65.264$

$b = -50.8742$

$y_1 \sim ax_1 + b$

STATISTICS

$r^2 = 0.9962$

$r = 0.9981$

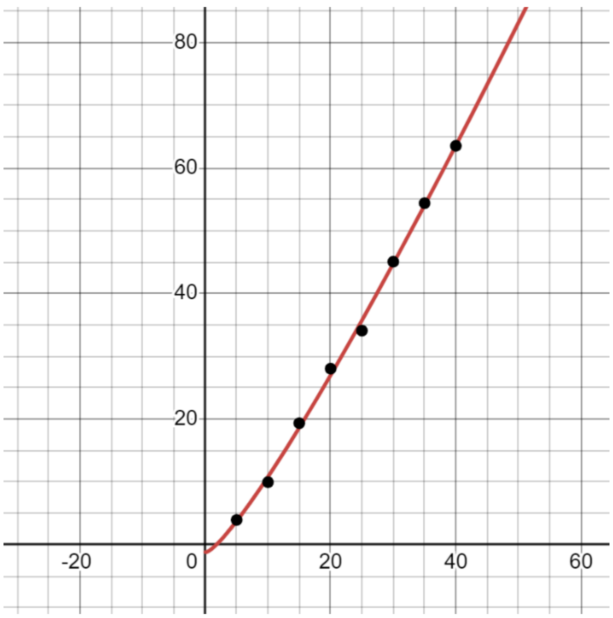
RESIDUALS

e_3 plot

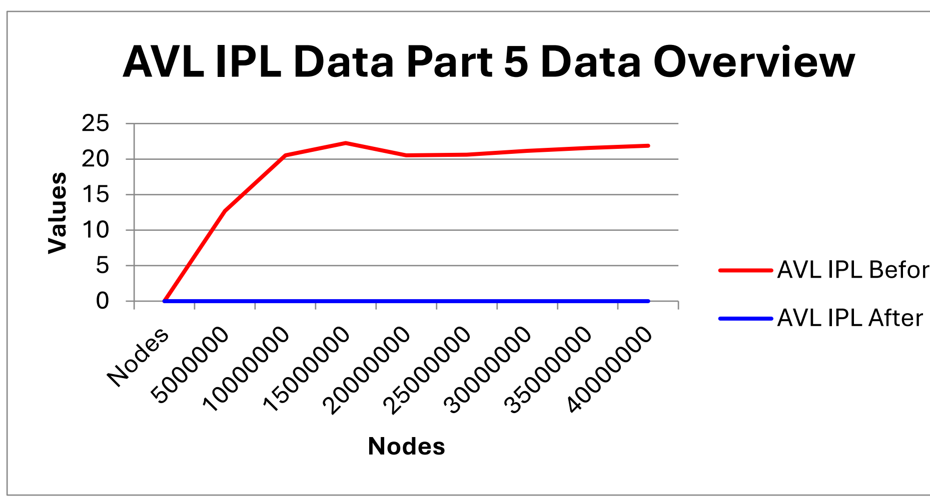
PARAMETERS

$a = 1.72121$

$b = -6.41201$



Nodes	AVL IPL Before	AVL IPL After
Nodes	AVL IPL Before	AVL IPL After
5000000	12.6793	NaN
10000000	20.5249	NaN
15000000	22.2545	NaN
20000000	20.5109	NaN
25000000	20.6074	NaN
30000000	21.1417	NaN
35000000	21.5704	NaN
40000000	21.8813	NaN



Red-Black Tree

$y_1 \sim a x_1 \cdot \log(x_1) + b x_1 + c$

STATISTICS

$R^2 = 0.999$

RESIDUALS

e_1 plot

PARAMETERS

$a = 2.95104$

$b = -2.85556$

$c = 7.4186$

$y_1 \sim a \cdot \log(x_1) + b$

Log Mode

STATISTICS

$R^2 = 0.827$

RESIDUALS

e_2 plot

PARAMETERS

$a = 84.6347$

$b = -70.8161$

$y_1 \sim a x_1 + b$

STATISTICS

$r^2 = 0.9788$

$r = 0.9893$

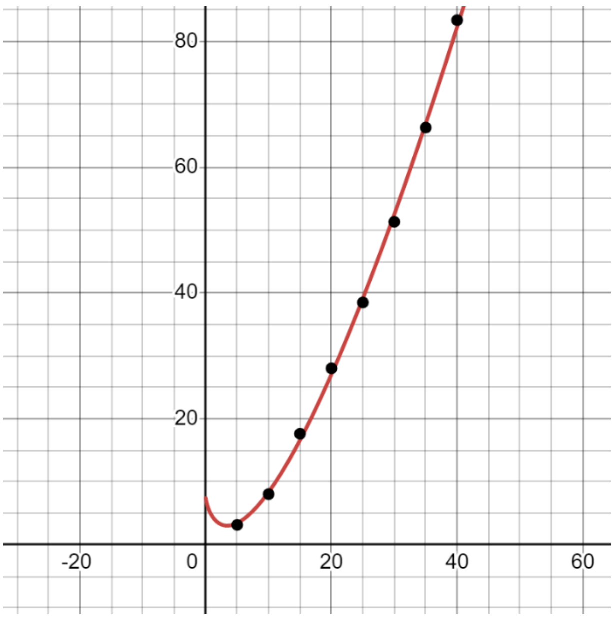
RESIDUALS

e_3 plot

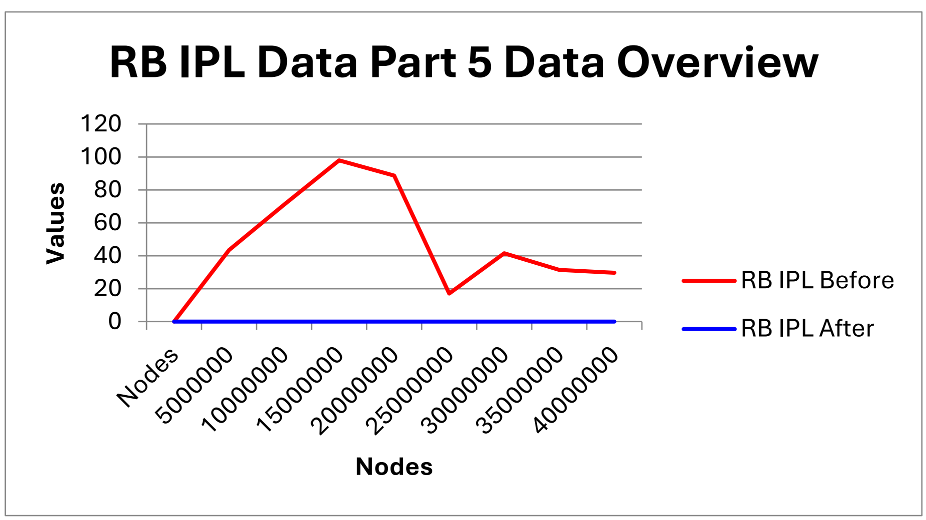
PARAMETERS

$a = 2.29659$

$b = -14.6089$



Nodes	RB IPL Before
Nodes	RB IPL Before
5000000	43.5206
10000000	71.0649
15000000	97.9021
20000000	88.7557
25000000	17.003
30000000	41.4261
35000000	31.4476
40000000	29.6145



CONCLUSIONS

Based on this data, we can answer multiple questions regarding AVLs and Red-Black Trees.

What is the time complexity of these trees?

In the context of this experiment (inserting, then running an insertion/deletion cycle), it can confidently be stated that the time complexity is $O(nlog(n))$, with every graph showing the closest correlation (r^2 value) to the equation $ax * \log(x) + bx + c$.

How are the average IPLs different?

The average Internal Path Lengths (IPL/n) show immense differences within the two algorithms. It can be observed that the average IPLs are almost always significantly lower in the AVL tree. This is because the specific implementation of AVLs used here includes a count property, where each node's count can be increased or decreased during insertion or deletion. This is especially applicable and apparent in samples with low data ranges. For example, in the second sample (Data Range = 50,000 & Cycles = Number of Nodes), the average IPL decreases as the number of nodes gets larger. This is because the tree isn't gaining nodes when duplicates are added or deleted, with the nodes already in the tree either increasing or decreasing their count instead. This immensely juxtaposes the Red-Black tree, of which does not have a count property. Instead, duplicate nodes are added to the tree as its own separate node, which dramatically increases the average IPL over time.

How does the Data Range affect these trees?

The data range has the greatest impact on the AVL tree, which, as discussed previously, utilizes a count property. This allows for a greatly reduced path when inserting a node or searching for a node to delete, especially as the data size gets larger. Therefore, when the data range is reduced, there are significant performance increases for the AVL trees. The Red-Black tree, however, is not nearly as affected. This can be clearly seen when comparing the first and second samples (Data Range = Nodes, Cycles = Nodes VS. Data Range = 0-50,000, Cycles = Nodes). Within these two samples, the AVL tree is consistently at least four times faster when limiting the data range. The Red-Black tree, however, sees negligible differences (In these samples, even performing worse!).

How does the number of cycles affect these trees?

Contrary to data range, the number of cycles has the greatest impact on the Red-Black tree. This can again be contributed to the count property, which greatly hinders the Red-Black tree strategy when searching and deleting a node. Since a Red-Black tree has significantly more nodes to search through, limiting the cycles allows for less searching and less tree transversals. Since increasing the cycles also increases the number of total operations to perform, it also has an adverse impact for the AVL tree. However, due to the count property, it does not nearly have as much of an impact, especially when the data range is set to a low number. A lower data range allows for the AVL to search less through the tree and allows for it to decrease more counts, which saves the process of having to perform rotations and rebalancing.

Which one is better? What are the use cases?

Although neither an AVL nor Red-Black tree is always superior, they both have distinct use cases. Firstly, an AVL is better when there is a constrained data range and/or when many duplicates are to be expected. This is due to the count property, which decreases both time and space complexity when duplicates are involved. Additionally, an AVL tree has a tighter balance property (and therefore a smaller maximum height), which aids in searching for a node, especially in larger datasets. In almost every case of this report, the AVL tree showcased shorter run times and a smaller average IPL. However, due to the nature of these methods, the real use case of Red-Black trees may have been masked. Red-Black trees could be better than AVL trees when minimal cycles are required and when the dataset is not relatively bound (i.e. a database that holds different user's information). Red-Black trees suffer from an increased height and increased decrease/search due to the lack of a count property. However, it could have a more efficient insertion time due to the nature of the algorithms. This is especially reflected in the last sample (Data Range = 0-Nodes, Cycles = 50,000), where it was notably faster than AVL up until about 6 million nodes (the count property may have overcome it at this point). Regardless, when choosing which tree to use, it is important to look at the data and expected use cases to make the best decision.