DSA Homework 3 Report

Submitted By: Diksha Prakash (dp978)

Question 1:

The code is the implementation of the basic symbol-table API that uses 2-3 trees that are not necessarily balanced as the underlying data structure.
Within the code, a ‘global’ variable i.e. ‘k’ (line no. 9) has been defined which can be altered according to the desired number of random inputs.
The output of the code is the in-order traversal of the tree.

Question 2:

The experiment is conducted on the tree which has been generated in the previous problem.
The idea of path calculation is that first we add the depths of all the nodes (or edges, whichever implementation we choose) to get the internal path length. Intuitively, the result is then divided by the tree size to obtain the average.
The average path length calculation can be computed using the formula: $P_N = N + P_{Left} + P_{Right}$ where, ‘N’ is the size of node ‘P’, ‘P_{Left}’ and ‘P_{Right}’ refer to its two subtrees.
I have considered two scenarios for average path length calculations: when we count red nodes & we don’t count the red nodes.

CASE 1: When we don’t count the red nodes, the average path length for random and ordered input is as under:

N	Average Path Length (Ordered)	Average Path Length (Random)
1	1.0	1.0
2	1.0	1.0
4	1.5	1.5
8	2.5	2.038750
16	4.5	2.66500
32	8.5	3.356250
64	16.5	4.143906
128	32.5	4.995859
256	64.5	5.723359
512	128.5	6.576699
1024	256.5	7.422119
2048	512.5	8.194238
4096	1024.5	9.020971
8192	2048.5	9.893995

The plot for the average path length in ordered and random case are as in Figure 1 and Figure 2 respectively. I have used the curve fitting option in 'Trendline' available in Excel to find the best possible fitted curve and the eventual equation for the generated plots.

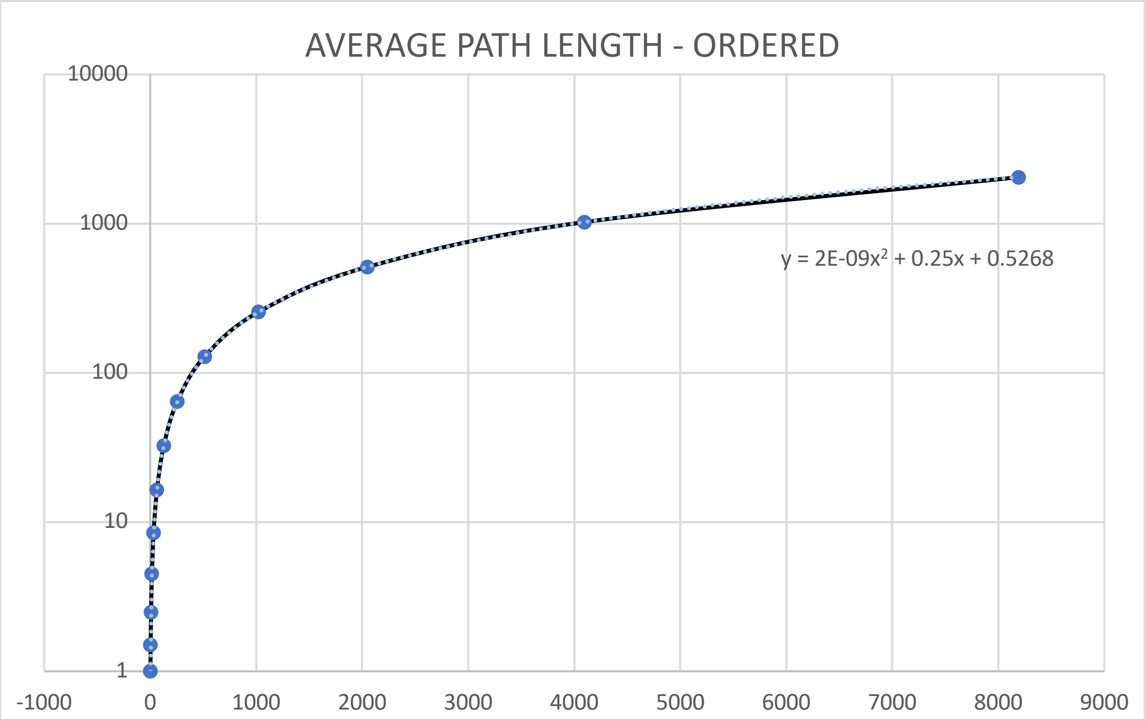

Figure 1: Plot for average path length calculation for set of ordered inputs.

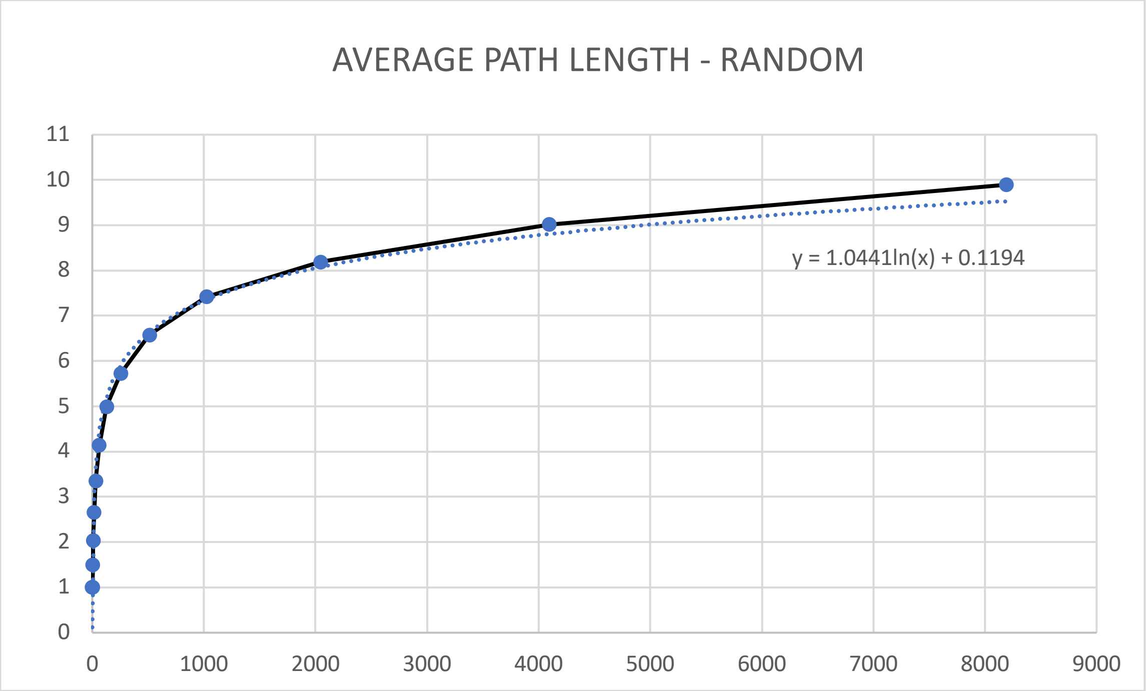

Figure 2: Plot for average path length calculation for set of random inputs.

The equation obtained for average path length in terms of the number of random inputs given is: $y = 1.0441ln(x) + 0.1194$
Whereas, the equation obtained for average path length in terms of the number of random inputs given is: $y = (2E - 09)x^2 + 0.25x + 0.5268$

CASE 2: When we count the red nodes, the average path length for random and ordered input is as under: (This is done by uncommenting lines 25 and 30 from the code Question2/Main.java while commenting lines 24 and 29. This could be automated using a 'switch' statement!)

N	Average Path Length (Ordered)	Average Path Length (Random)
1	1.0	1.0
2	1.5	1.5
4	2.5	2.2475
8	4.5	3.1825
16	8.5	4.190625
32	16.5	5.4071875
64	32.5	6.64125
128	64.5	7.9759375
256	128.5	9.145390625
512	256.5	10.65126953125
1024	512.5	12.107880859375
2048	1024.5	13.444296875
4096	2048.5	14.7584033203125
8192	4096.5	16.13838623046875

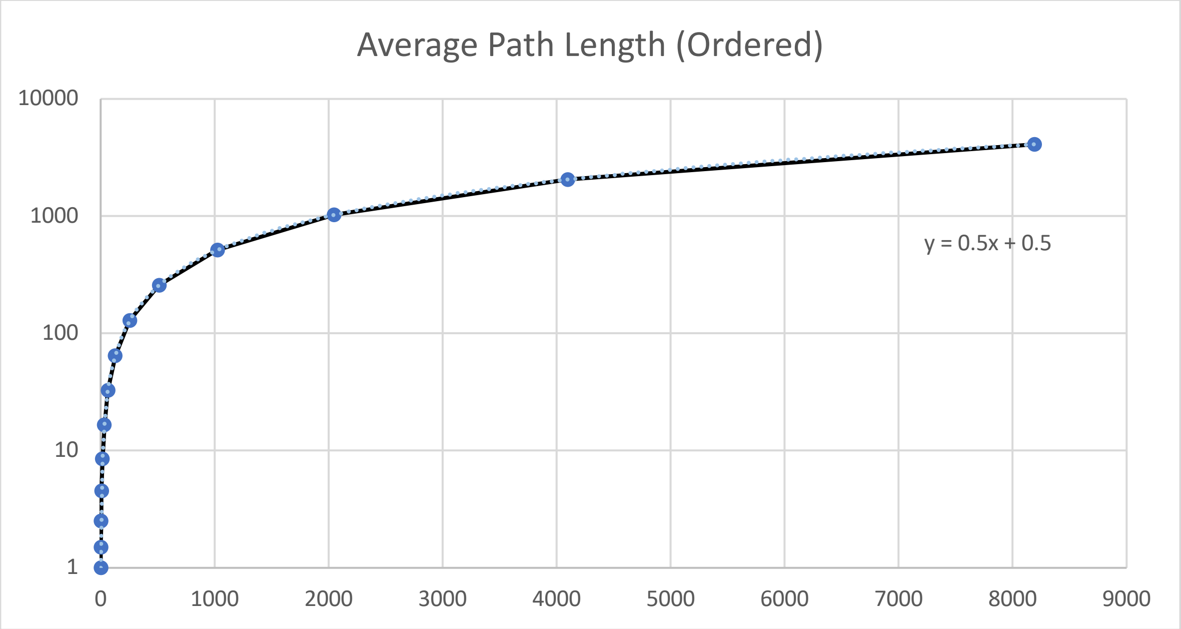

Figure 3: Plot for average path length calculation for set of ordered inputs.

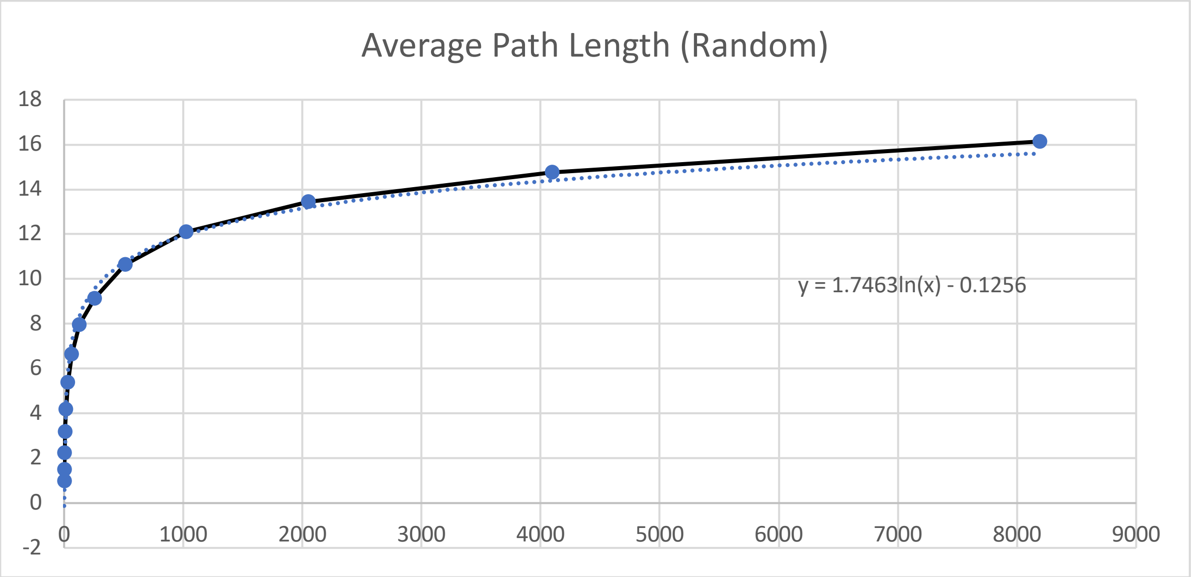

Figure 4: Plot for average path length calculation for set of random inputs.

The equation obtained for average path length in terms of the number of random inputs given is: $y \ = \ 1.7463n(x) - \ 0.1256$

Whereas, the equation obtained for average path length in terms of the number of random inputs given is: $y \ = 0.5x \ + \ 0.5$

<u>Question 3:</u>

The results obtained are as follows:

> **"The result of 10000 nodes is: 25.4033%**
> **The result of 100000 nodes is: 25.3931%**
> **The result of 1000000 nodes is: 25.3883%"**

Thus, the average percentage of nodes in a random input 'Red-Black Tree' is approximately 25.4%

<u>Question4:</u>

Here, the calculation for the 'internal' path length is the same as in Question #2. Mathematically, the internal path length is represented as: $P_N = N + \ P_{Left} + P_{Right}$ where, 'N' is the size of node 'P', 'P_{Left}'' and 'P_{Right}'' refer to its two subtrees.

The plot for the average length as obtained in Excel is as under:

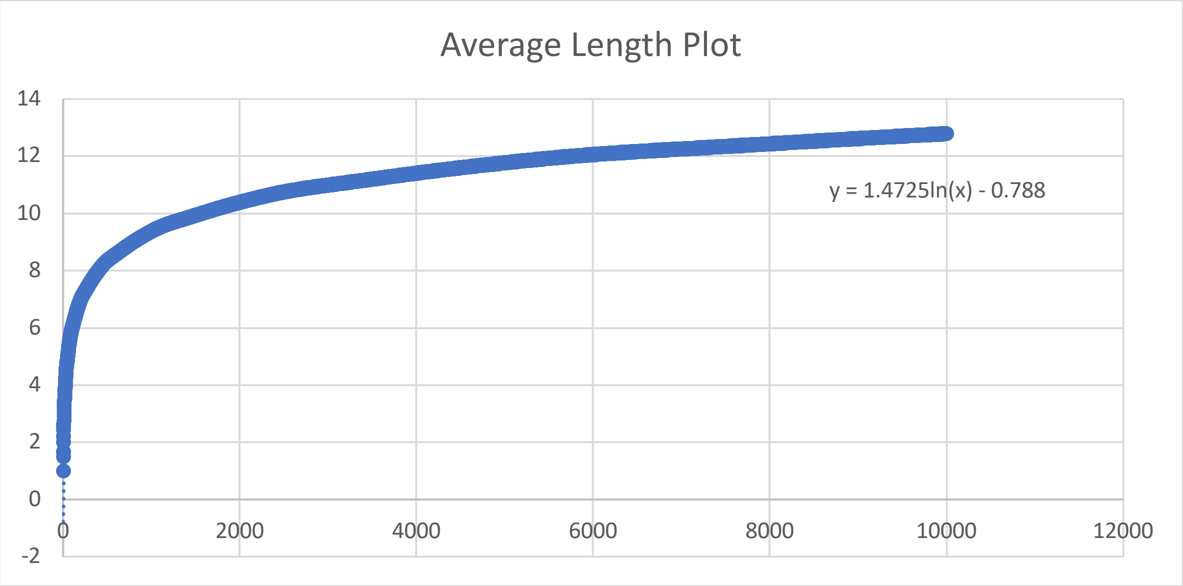

Figure 5: Plot for average path length.

From the curve fitting the equation obtained for average path length in terms of the number of inputs given is: $y = 1.4725ln(x) - 0.788$
The computed standard deviation is quite small, and the average value is 0.070

The code took almost 4 hours to run on my laptop with the specs: 1.8 GHz Intel Core i5 to work all the results.

Question 5:

The output as produced by the code is:

"Result of Select(7) is : 8
Result of Rank(7) is : 6"