Data Structure Hw3

Name: Xiaoyi Tang
NetID: xt59

Q1.

- Code: TwoThreeTree.java
- API:

public class ST<Key, Value>	
ST ()	Create a Symbol Table
void put (Key key, Value val)	put key-value pair into the table
value get (Key key)	value paired with key
void delete (Key key)	remove key (and its value) from table
boolean contains (Key key)	Find whether there is a value paired with key
booean isEmpty ()	Return whether the table is empty
int size ()	number of key-value pairs in the table
Iterable<Key> keys ()	all the keys in the table

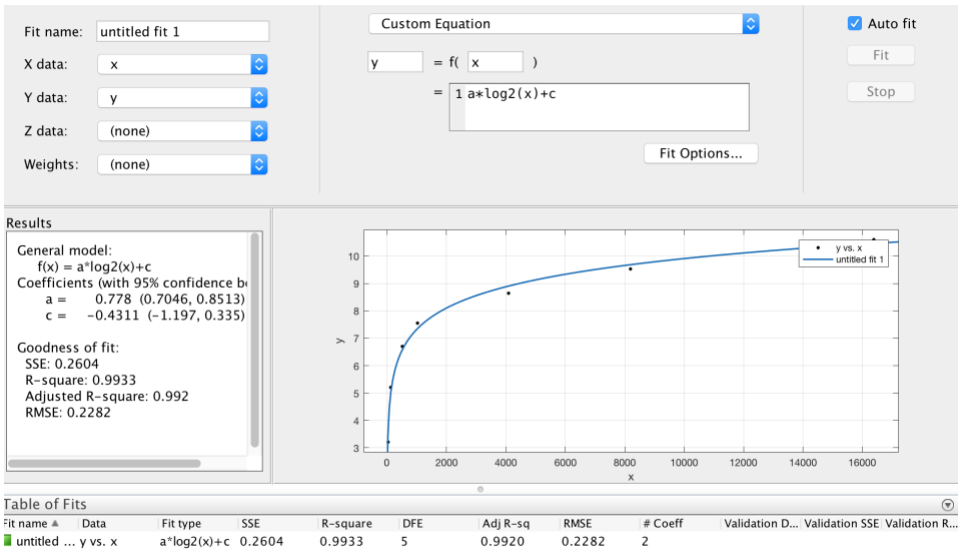
Q2.

- Code: TTT.java
- Hypothesis:

Size(N)	Path Length (random insertion)	Path Length (sorted insertion)
32	3.22	8.21
128	5.21	30.49
512	6.7	129.77
1024	7.55	255.8
4096	8.65	955.65
8192	9.52	1970.42
16384	10.59	3997.55

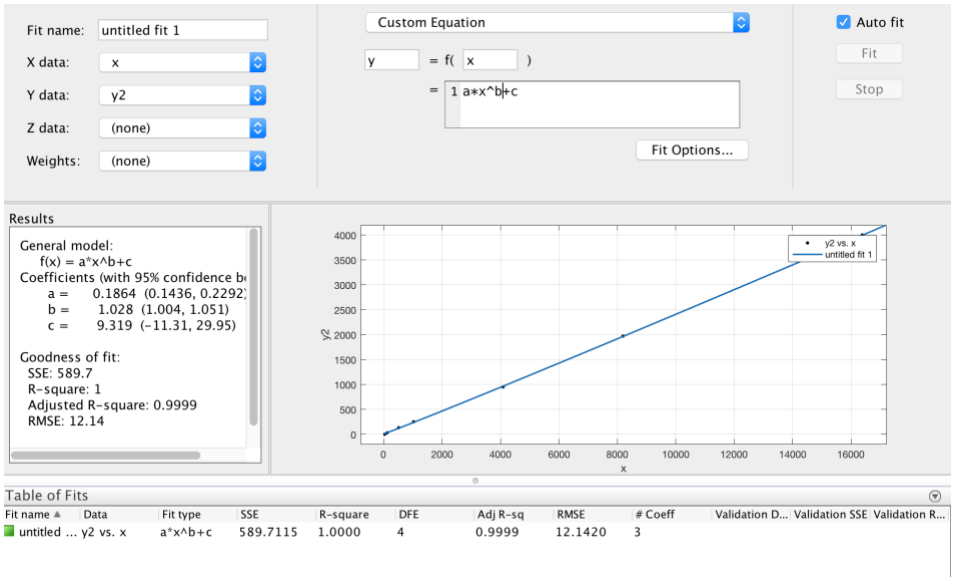
To get the hypothesis, I used the cftool in Matlab2016 to get the results.

- Random insertion:



For path length in Red-Black BST, this case can be considered as average case, it can be assumed to satisfy the equation $a * \log_2 N + b$, so the average path length for random insertion is $P(N) = 0.778 * \log_2 N - 0.431$

- Sorted insertion:



this case can be considered as worst case in inserting node into not balanced 2-3 tree. For the data curve fitting, we can get the suitable fit: $P(N) = 0.186 * N^{1.028} + 9.319$, it is almost linear!

Q3.

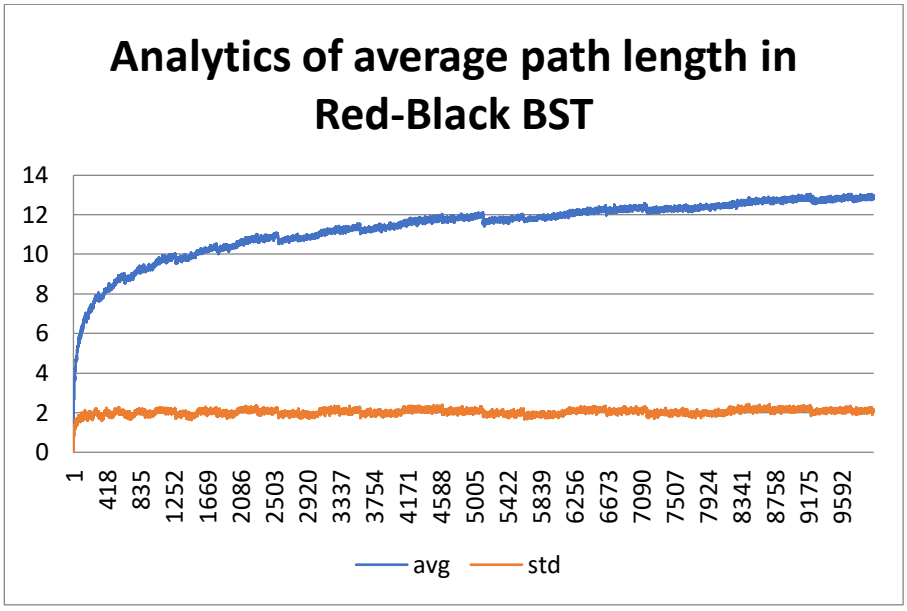
- Code: RBTree.java
- Hypothesis:

Size (N)	Percentage
10^4	0.2467
10^5	0.2451
10^6	0.24085

It is obvious that for $N = 10^4, 10^5, 10^6$, the percentage of red nodes is around 25%, so we can get the hypothesis that: the percentage of red nodes in Red-Black Tree is about 25%.

Q4.

- Code: RBT.java
- Results:
  In my code, I saved all data results into a csv file and got a graph based on that.



Q5.

- Code: BST.java
- The value of rank (7) is 6
- The value of select (7) is 8