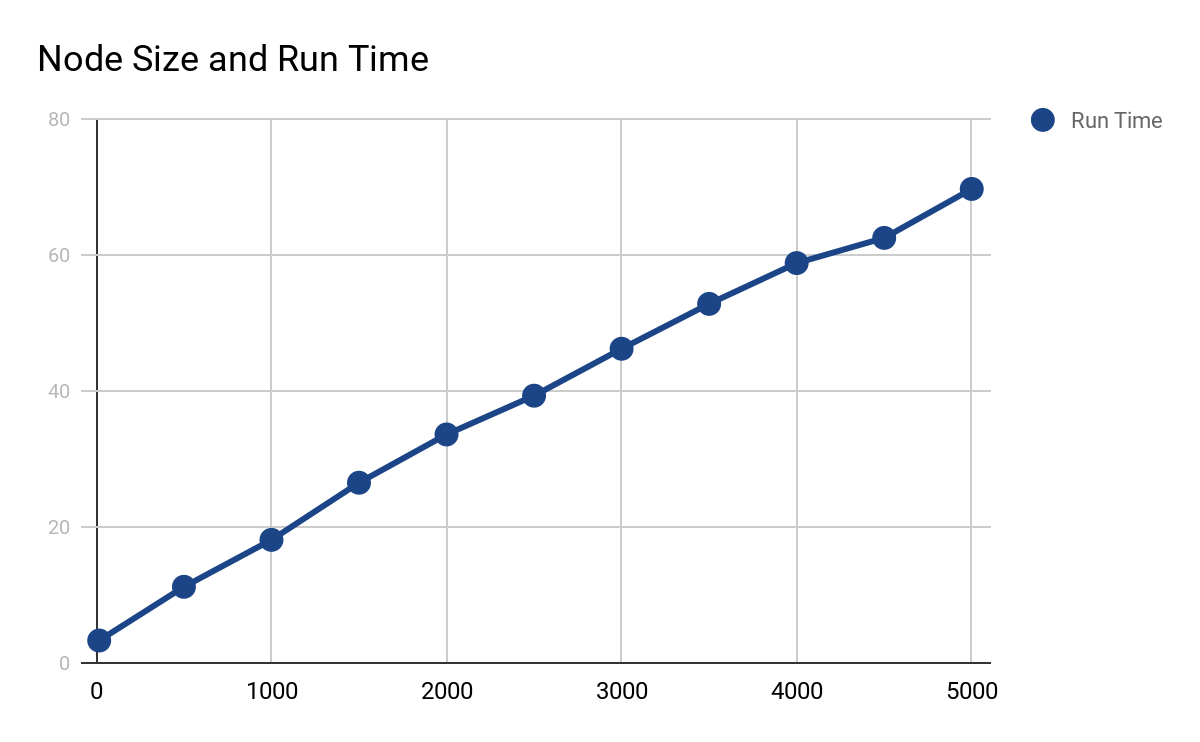
Name: Jordan Cheung

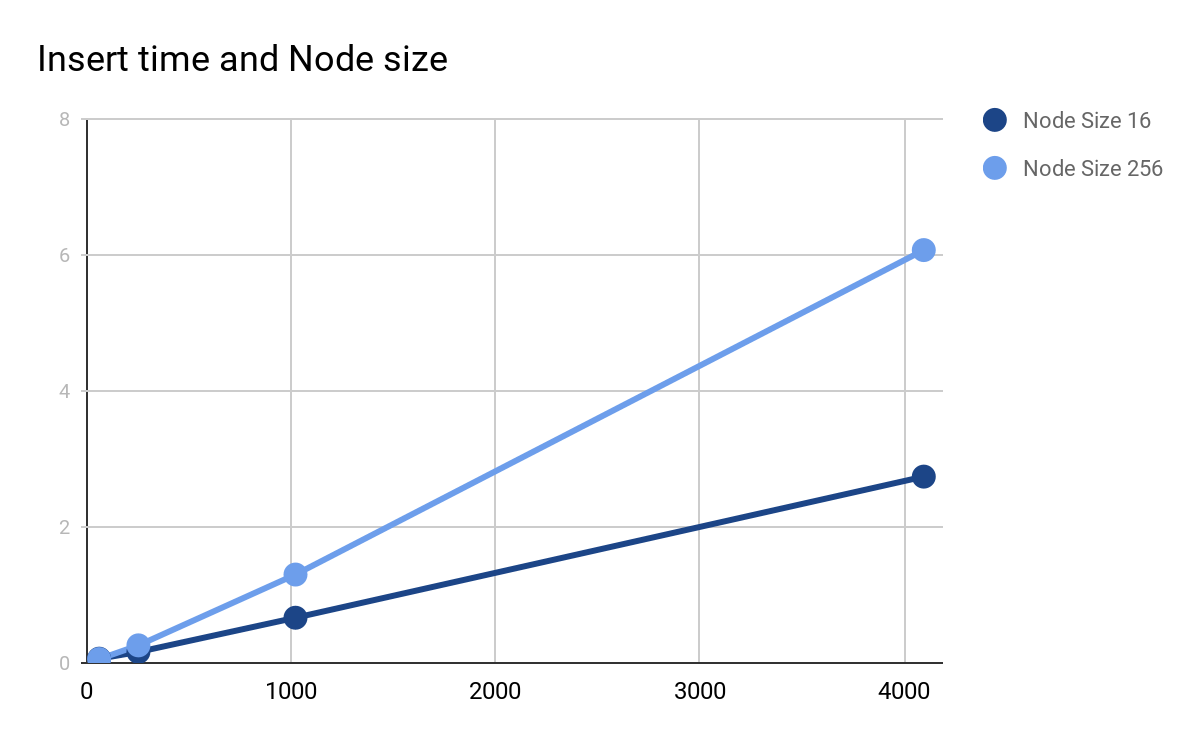
Student Number: 301320290

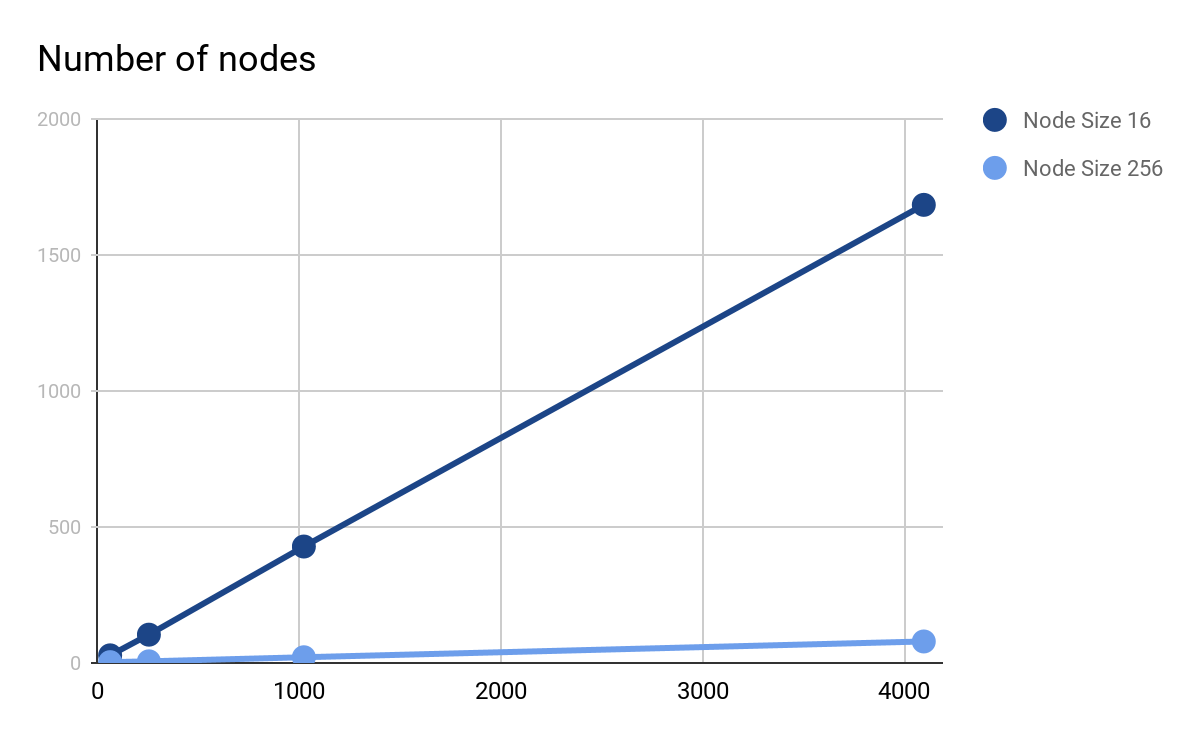
Experiment 1:

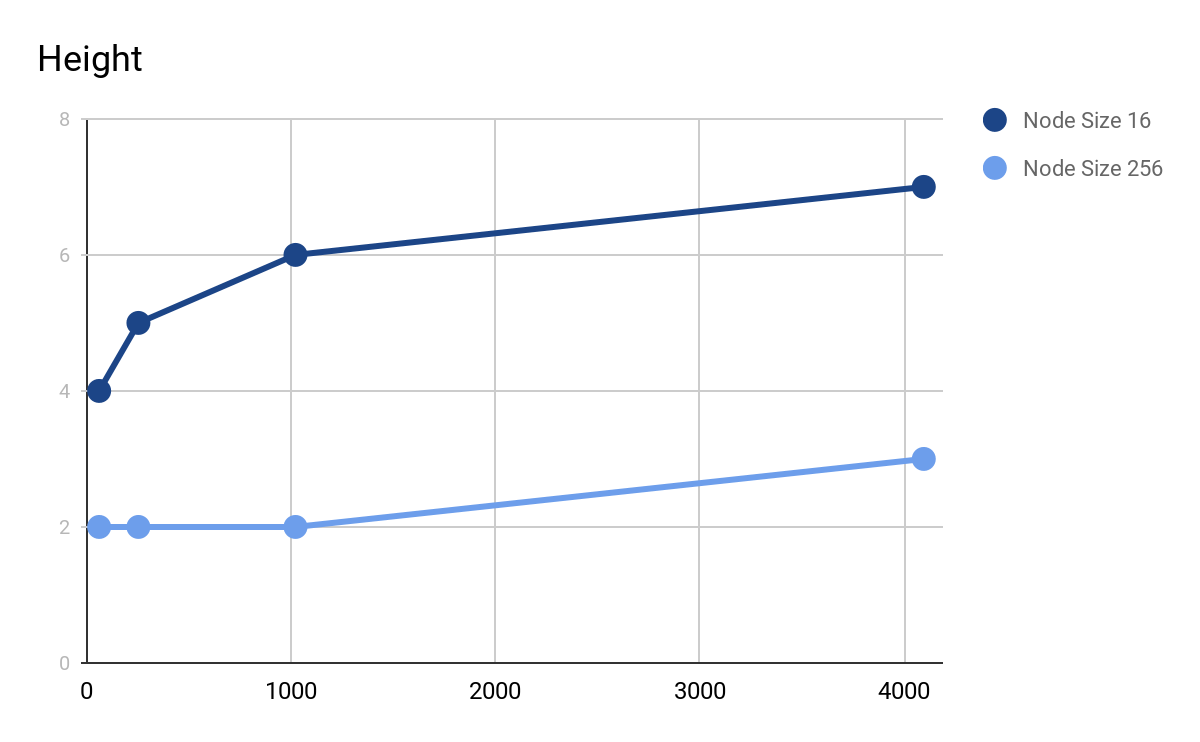


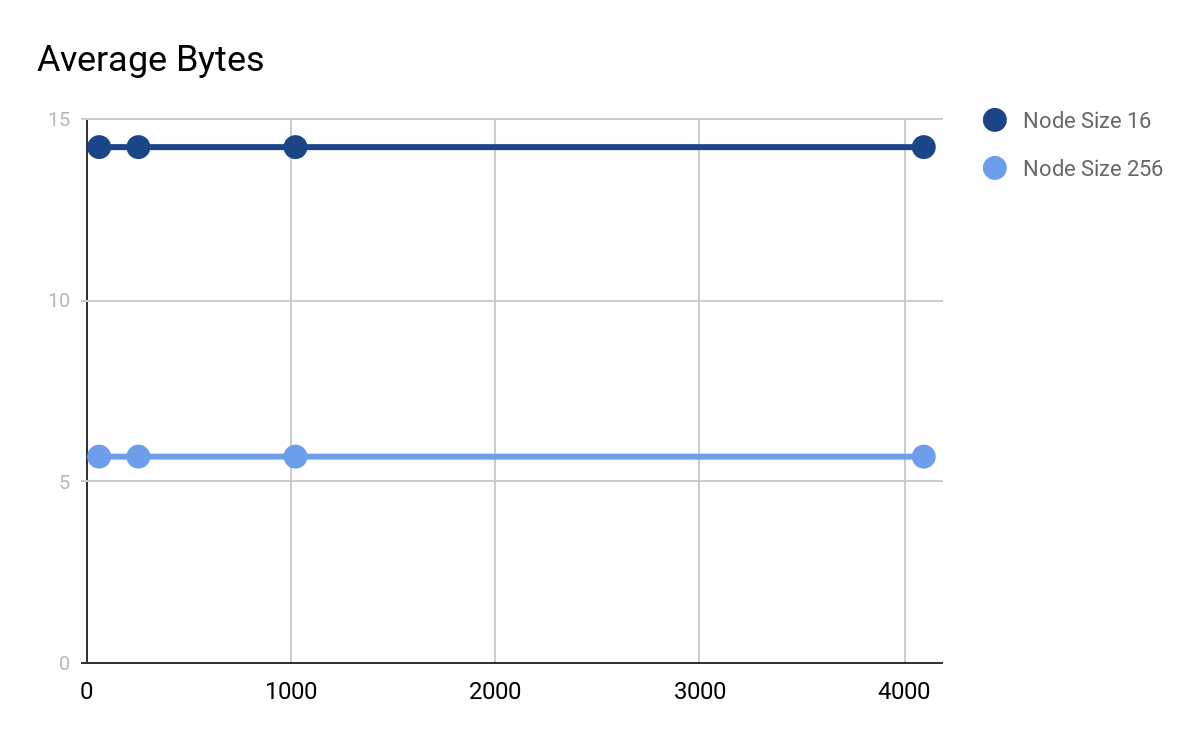
For this experiment i chose to use the int data type which is the size of 4 bytes. I chose to insert 5000 keys randomly using the rand() function and the possible values for the keys were from 1 to 50000. The node sizes that I chose to test was from starting at node size of 16 and i incremented by 500 each time until 5000 which was the number of keys i chose. After plotting the data values on the graph, it appears that as you increase the node size, the run time increases in a linear rate.

Experiment 2:









For the node sizes I picked 16 bytes which is the lowest possible for the int data type and I picked 256 because it was the default node size for the program. I chose to insert the numbers 64, 256,1024 and 4096. The reason why I chose to insert 64 was because 64 is exactly where the height of the tree with a node size 256 went from having a height of 1 to a height of 2. This is because if you divide the node size of 256 by the size of int which is 4 you get 64 which means that each node can hold up to 64 keys. For the other values of N I chose to multiply 64 by 4 multiple times because 16 and 256 can fit in easily in 256, 1024, 4096.

The insert time for a node size of 256 was significantly longer than the insert time of node size 16. The number of nodes was heavily correlated with the number of keys inserted. For every time i multiplied the number of keys I inserted, the number of nodes would also increase by 4 times. Node size 256 had significantly less nodes than node size 16 because each node in Node size 256 could hold up to 64 keys while node size 16 could only hold up to 4.

Height was heavily correlated with the number of keys. I found that height was just log64(number of keys) + 1 and log4(number of keys) + 1. I got log4 because for 16/4 gives the number of keys for each node. Using the formula of log4(number of keys) + 1, I was able to recreate the heights that I found when running the experiment. Log4(64) + 1 = 4, Log4(256) + 1 = 5, Log4(1024) + 1 = 6 and Log4(4096) + 1 = 7. The +1 comes from the root node counting as height 1. What is happening is that at the values 64, 256, 1024 and 4096, all of the nodes are completely full and the height is being forced to increase.

For the average bytes it remained the same all values of N I tried. So therefore the average bytes is constant no matter what number of N is inserted.