Jing Wen
301327176

Experiment one

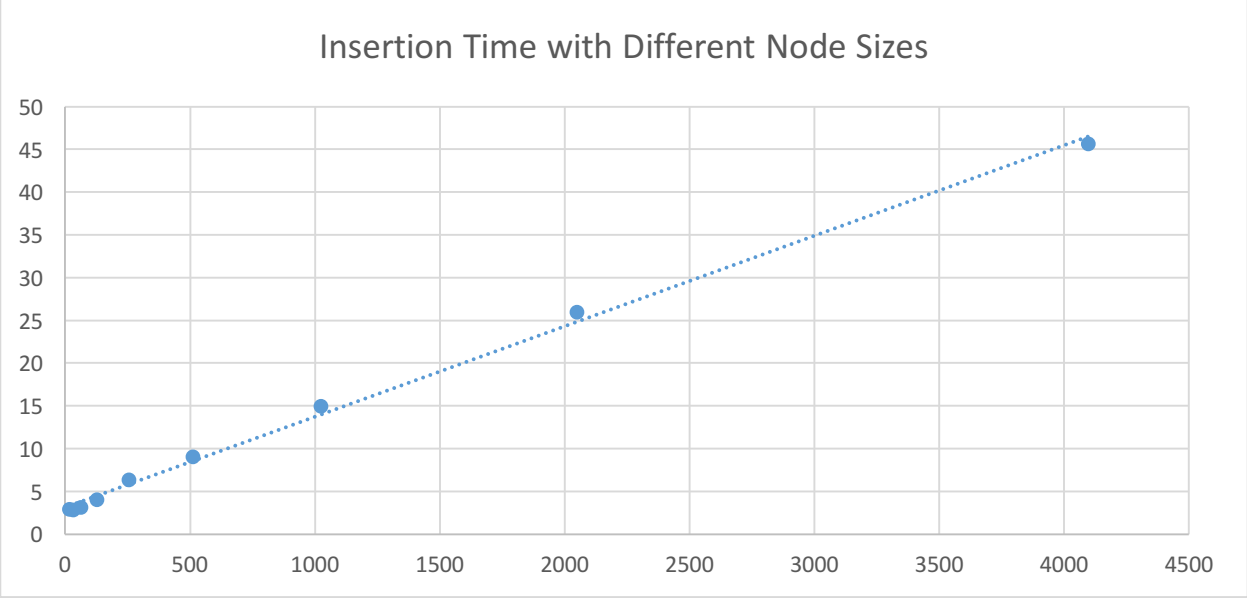
Brief Introduction

Figure 1 shows the insertion time with different node sizes.
The experiment result is achieved under the above condition:
Data type choosing: Integer (int);
Node sizes: 16, 32, 64, 128, 256, 512, 1024, 2048, 4096;
Inserted array size (or N): 5000

Table 1

Node Size	tier 1	tier 2	tier 3	tier4	tier 5	Average
16	2.888	2.851	2.948	2.801	3.005	2.8986
32	2.91	2.8	2.812	2.736	2.893	2.8302
64	3.064	3.144	3.204	3.12	3.055	3.1174
128	3.966	3.999	3.979	3.931	4.05	3.985
256	6.039	6.397	6.047	6.32	6.731	6.3068
512	9.229	8.809	9.205	8.823	8.944	9.002
1024	15.16	15.019	14.87	14.528	15.067	14.9288
2048	25.882	26.092	26.427	26.234	25.224	25.9718
4096	46.016	45.14	45.529	45.384	46.046	45.623

Figure 1



Jing Wen
301327176

Conclusion

When experimenting, one important thing to notice is that the minimum node size for data type int is 16. From the above chart, we can see that, as node sizes increase, the time spend to insert also increase, despite there is a little drop down when node size is 32. Here is my guess if we keep increasing the node size, the line will look more like a logn curve. The experiment also shows the importance of choosing right node sizes; large node size may waste time and hurt efficiency.

Jing Wen
301327176

Experiment Two

Brief Introduction

Figure 2 to Figure 5 show nodes, heights, average-number-per-data, and insertion time with different node sizes respectively.
The experiment result is achieved under the above condition:
Data type choosing: Integer (int);
Node sizes: 16, 64, 1024;
Inserted array size (or N): 2000, 5000, 8000, 12000

Talbe2

N = 2000	nodes	height	average_bytes_per_data	time
16	542	6	14.2222	1.804
64	126	3	7.11111	2.216
256	28	2	5.68889	2.584
1024	7	2	5.41799	5.488
N = 5000				
16	1370	7	14.2222	4.649
64	321	4	7.11111	4.351
256	58	2	5.68889	6.665
1024	16	2	5.41799	14.504
N = 8000				
16	2188	7	14.2222	6.817
64	518	4	7.11111	6.884
256	98	3	5.68889	11.013
1024	23	2	5.41799	24.523
N = 12000				
16	3269	8	14.2222	10.708
64	763	4	7.11111	9.999
256	149	3	5.68889	16.537
1024	35	2	5.41799	36.323

Jing Wen
301327176

Figure 2

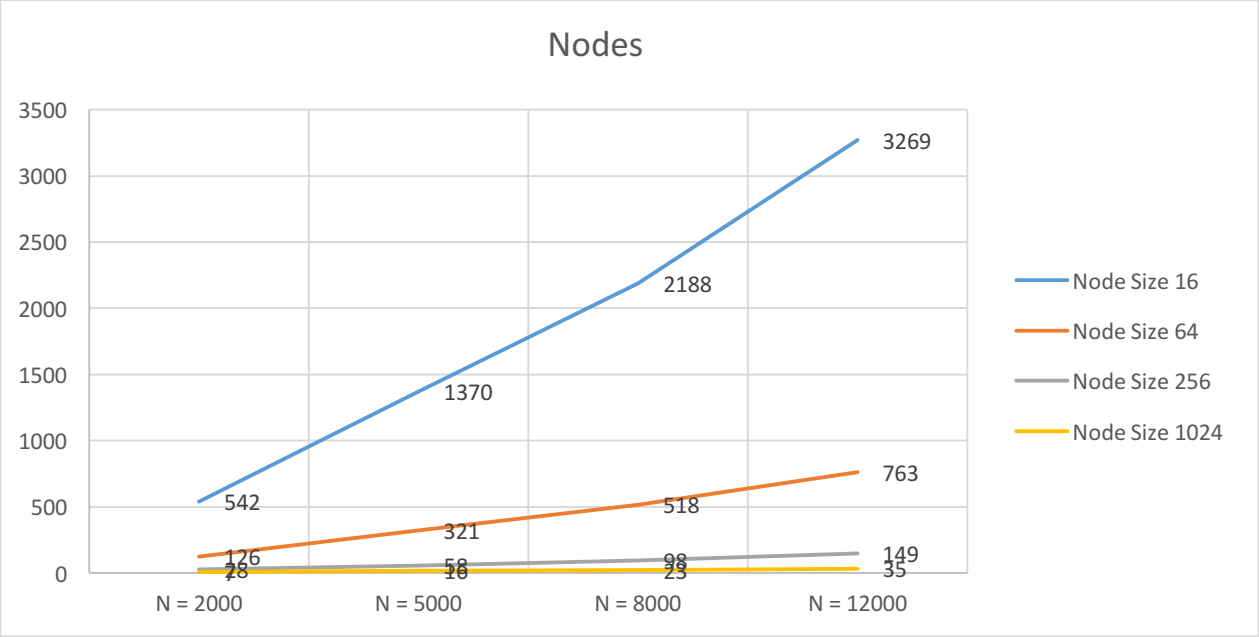
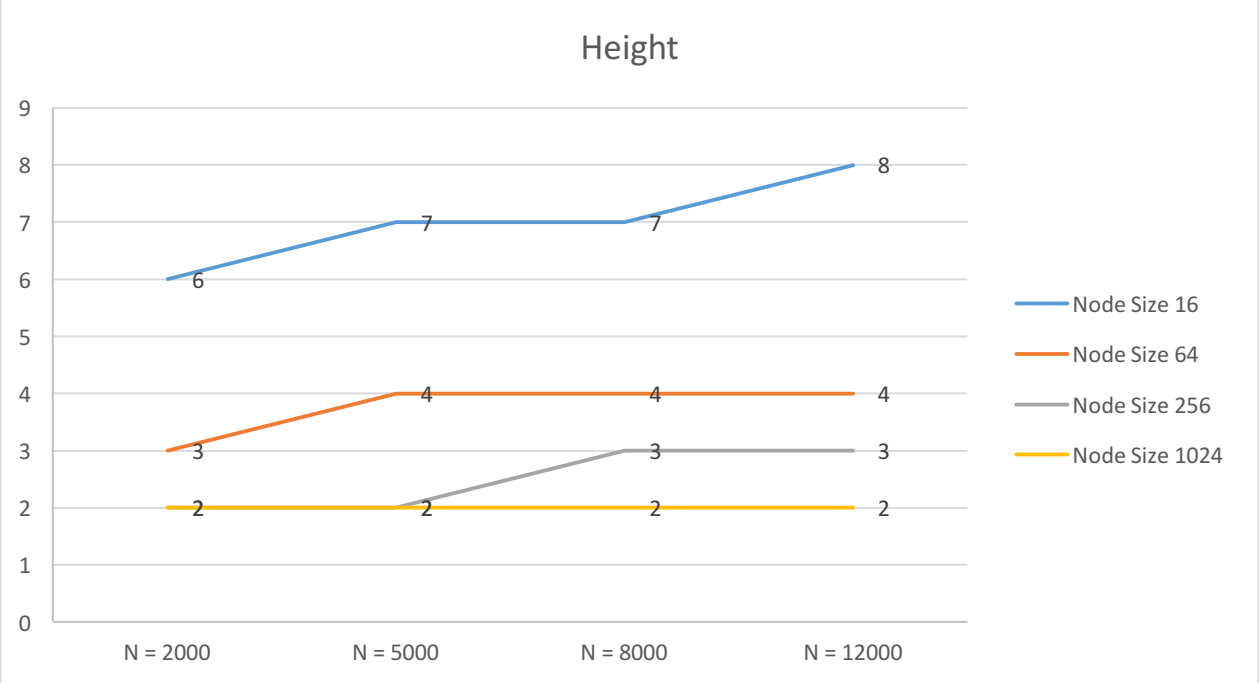


Figure 3



Jing Wen
301327176

Figure 4

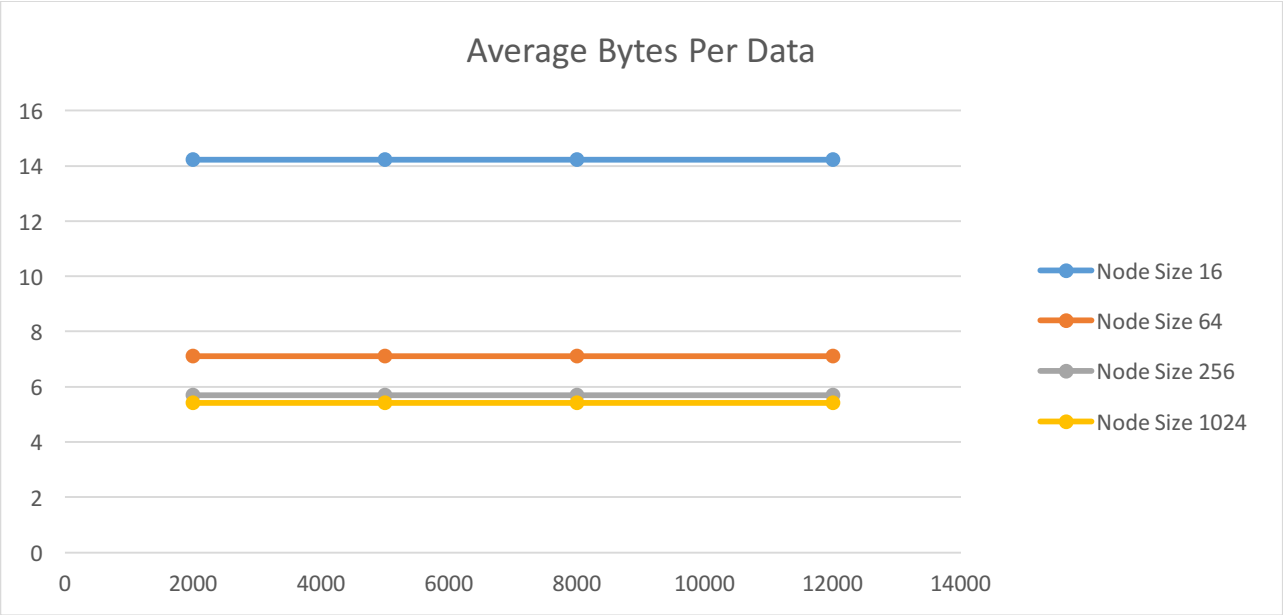
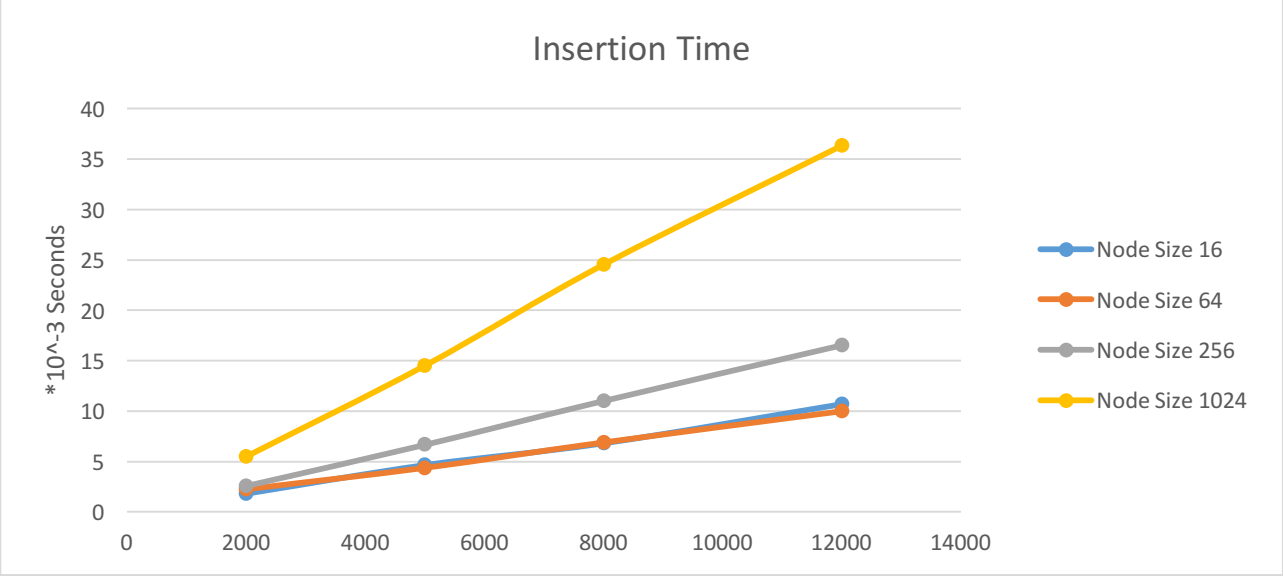


Figure 5



Jing Wen
301327176

Conclusion

From the above figures, we can see that the greater the node size is, the smaller height, node, average-byte-per-data will have; but greater insertion time, despite when node size is 64 which is almost slightly less than node size is 16. Another fact is that the average-byte-per-data will always stay the same if the node size does not change, however the insert array size increase or decrease. Again, it is important to choose right node size.