Test Cases: Eclips

Test

```
連 eclipse-workspace - assignment5/src/edu/neu/coe/info6205/bqs/Queue_Elements.java - Eclipse IDE
              🖺 Package Explorer 😈 JUnit 🕱 🖟 🔐 🚨 🔝 🕵 🦺 🔳 🗒 🔻 🔻 🗖 🔝 BSTSimple.java
                                                                                                                     Element secondNewest = newest;
if (isEmpty()) oldest = element;
Finished after 0.035 seconds
                                                                                                                     assert secondNewest != null; // Redundant Check
secondNewest.next = element;
                                                                                            32
33
34
35
36
37
38
▼  edu.neu.coe.info6205.symbolTable.BSTTest [Runner: JUnit 4] (0.001 s)
                                                                                                                this.newest = element;
     testPut0 (0.000 s)
      testPut1 (0.000 s)
      testPut2 (0.000 s)
     testPut3 (0.000 s)
                                                                                                           * Dequeue an element from the oldest list and return the item.
     testPutN (0.000 s)
                                                                                             39

40

41

42 ⊕

43

44

45

46

47

48

49

50

51

52
                                                                                                            @return the value of the oldest element.
      testDelete1 (0.000 s)
     testDelete2 (0.000 s)
                                                                                                         public Item dequeue() {
   if (isEmpty()) return null;
     testDelete3 (0.000 s)
     testDelete4 (0.000 s)
     testPutAll (0.000 s)
                                                                                                                   assert oldest != null; // Redundant assertion
Item result = oldest.item;
oldest = oldest.next;
if (isEmpty()) newest = null;
     testTraverse (0.000 s)
      testSetRoot1 (0.000 s)
                                                                                                                    return result;
}
     testSetRoot2 (0.000 s)
                                                                               →
Failure Trace
                                                                                             52
53
54
55
56
57
58
                                                                                                         public boolean isEmpty() {
    return oldest == null;
                                                                                                         // This Element essentially begins a LinkedList of Elements which correspond
// to the elements that can be taken from the queue (head points to the oldest elemen
// However, it is built in manner that requires a pointer to the newest element.
```

In this assignment, I ran the experiment for different number of nodes: 100, 1100, 2100, 3100, 4100, 5100, 6100, 7100, 8100, 9100, 11000 Adjust number of operation (number of inserts/deletions) and Key range for each experiments.

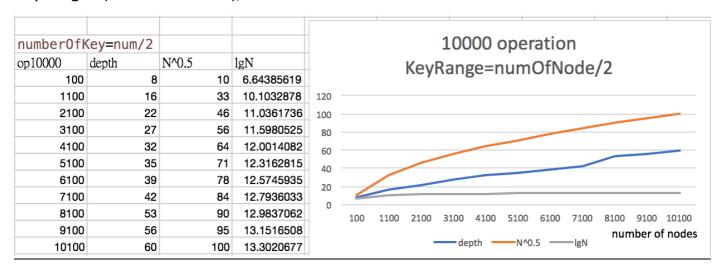
1. Experiment Result:

Some useful abbreviations:

- N Number of nodes.
- depth The largest depth of tree.

Experiment 1: 10000 operations

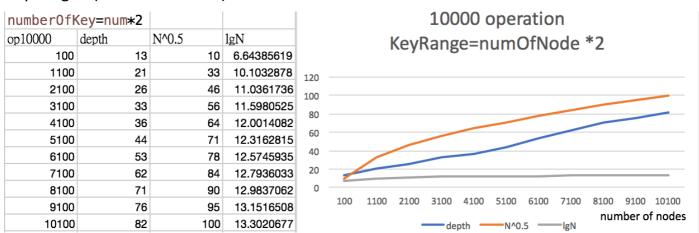
Key range= (number of node)/2



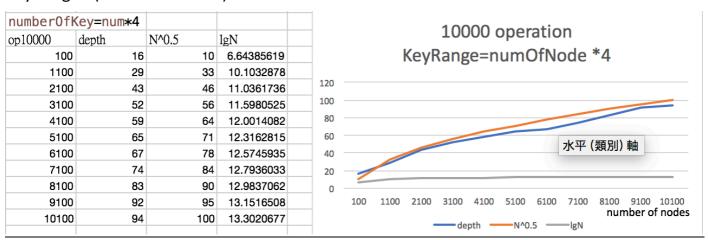
Key range= number of node

numberOf	Key=num														
op10000	depth	N^0.5	lgN	10000 operation											
100	10	10	6.64385619												
1100	19	33	10.1032878	KeyRange=numOfNode											
2100	24	46	11.0361736	120											
3100	27	56	11.5980525												
4100	34	64	12.0014082	100											
5100	44	71	12.3162815	80											
6100	43	78	12.5745935	60											
7100	46	84	12.7936033	40											
8100	49	90	12.9837062	20											
9100	55	95	13.1516508	0											
10100	60	100	13.3020677		100	1100	2100	3100	4100	5100	6100	7100	8100	9100	
							-	de	pth —	N^0	.5 —	—lgN	nun	nber o	f nodes

Key range= (number of node)*2



Key range= (number of node)*4



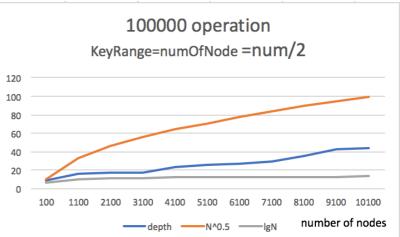
In this experiment, we can conclude that when key range is small, the largest depth of tree will close to IgN.

When the key range is more than twice of number of node, the depth line will close to N^0.5.

Experiment 2: 100000 operations

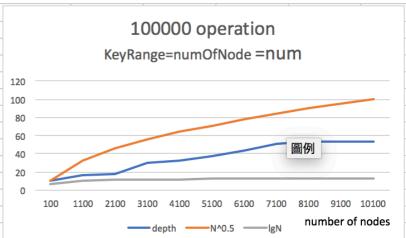
Key range= (number of node)/2

number0fk	(ey=num/2				
op100000	depth	N^0.5	lgN		
100	9	10	6.64385619		
1100	16	33	10.1032878		
2100	17	46	11.0361736		
3100	18	56	11.5980525		
4100	23	64	12.0014082		
5100	26	71	12.3162815		
6100	27	78	12.5745935		
7100	29	84	12.7936033		
8100	35	90	12.9837062		
9100	43	95	13.1516508		
10100	44	100	13.3020677		



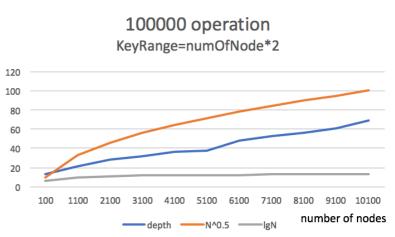
Key range= number of node

number0fk	(ey=num		
op100000 depth		N^0.5	lgN
100	10	10	6.64385619
1100	17	33	10.1032878
2100	18	46	11.0361736
3100	30	56	11.5980525
4100	32	64	12.0014082
5100	38	71	12.3162815
6100	43	78	12.5745935
7100	51	84	12.7936033
8100	54	90	12.9837062
9100	54	95	13.1516508
10100	54	100	13.3020677

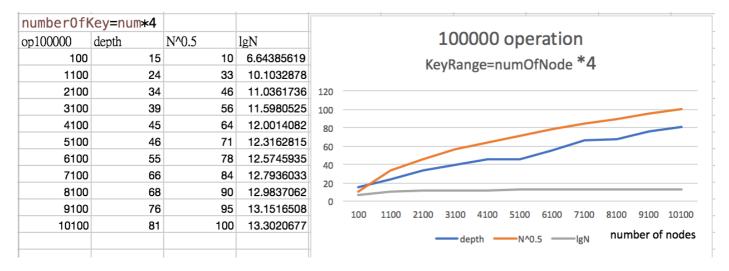


Key range= (number of node)*2

	•		
number0fk	(ey=num∗2		
op100000	depth	N^0.5	lgN
100	13	10	6.64385619
1100	21	33	10.1032878
2100	28	46	11.0361736
3100	32	56	11.5980525
4100	37	64	12.0014082
5100	38	71	12.3162815
6100	48	78	12.5745935
7100	53	84	12.7936033
8100	56	90	12.9837062
9100	61	95	13.1516508
10100	69	100	13.3020677



Key range= (number of node)*4



The same trend with experiment 1.

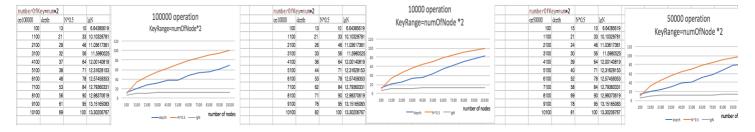
In this experiment, we can still also conclude that when key range is small, the largest depth of tree will close to IgN.

When the key range is more than twice of number of node, the depth line will close to N^0.5.

When we did more operation, the depth will smaller than the depth of less operation, that mean the depth line will more close to IgN. (compare with less operation)

Experiment 3: Compare different operations

We assume that Key range are all equal (number of node)*2.



In this experiment, we compare different operation, we can see that the more operations(100000) we did, the depth line will be more close to IgN.

When we did 1000 operation and 5000 operations, the depth line almost same, that means more close to N^0.5.

Conclusion:

When operations is 100000, the depth line will more close to IgN. (compare to 10000 operations) When the operations is less than 100000 and above 10000, the depth line will more close to N^0.5. When the Key range is relatively small, the depth line will close to IgN, and when we increase the Key range, the depth line will stay away from IgN that mean close to N^0.5.