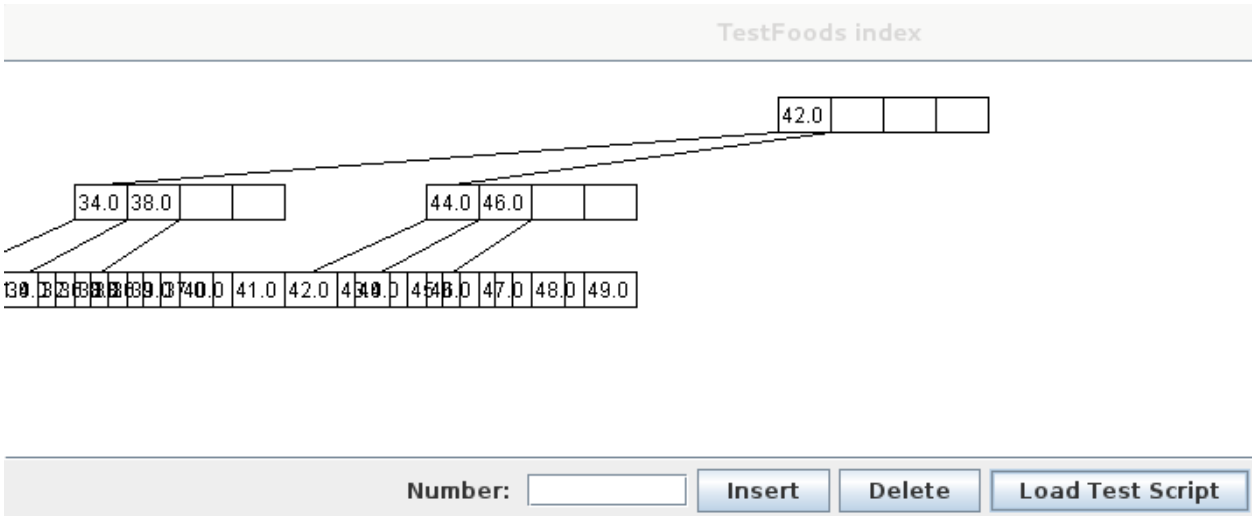
for d=2;

Simple test:

Insert:
[java] m_readNumDiskOps:319
[java] m_writeNumDiskOps:135

we get the inserted graph like below:



Delete:
[java] m_readNumDiskOps:520
[java] m_writeNumDiskOps:135

Clean:
[java] m_readNumDiskOps:681
[java] m_writeNumDiskOps:136

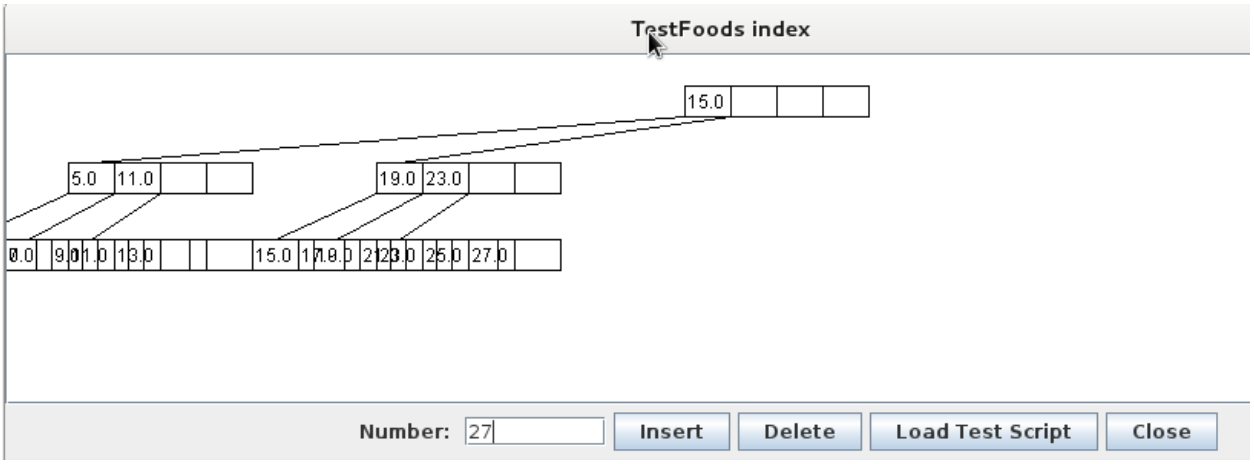
as the desired nodes do not exist in the tree, the delete method only read and scan the data,namely only adds the Num of m_readNumDiskOps.

Testing scripts:

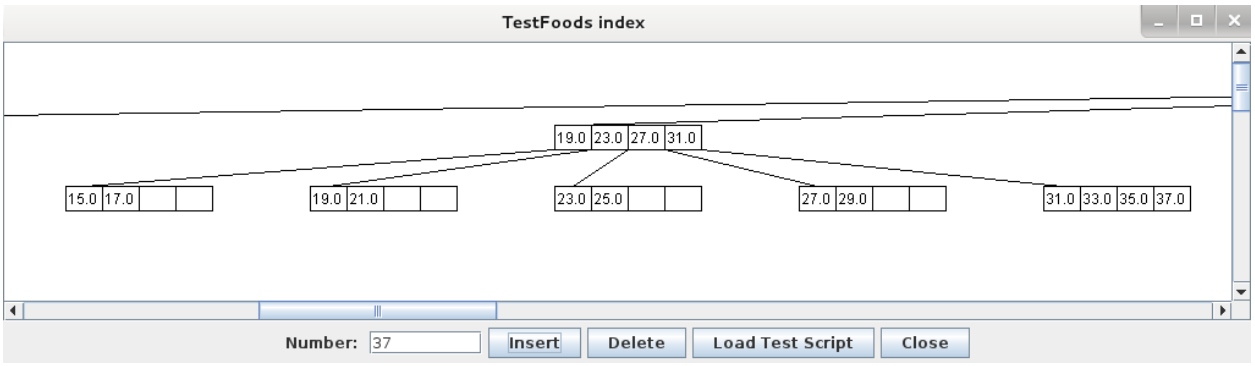
Here we use positive numbers as insert method,and negative numbers for delete method:
1 3 5 7 9 4 -3 -5 11 13 15 17 19 23 21 25 27 //picture1
29 31 33 35 37 //picture2
-27 (for myscript1,this is the end,myscript skips this key)//picture5
39//my picture3

-35//my picture4
41 43 -43()

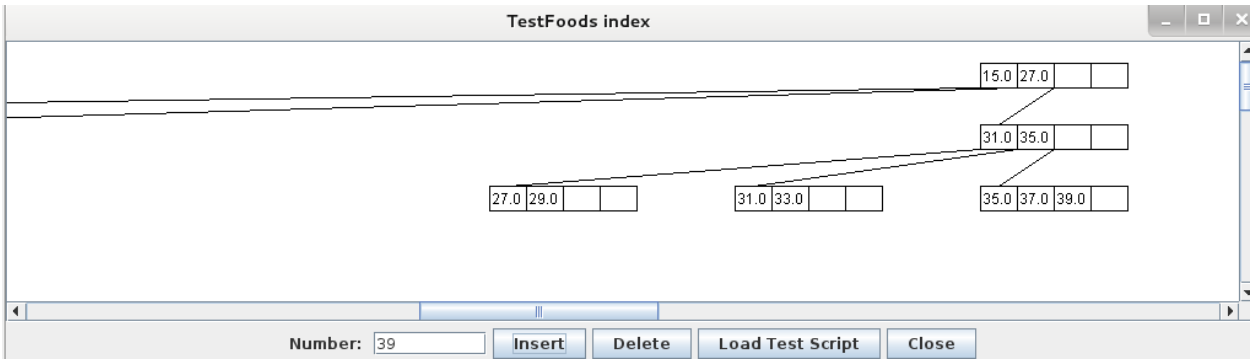
picture1:



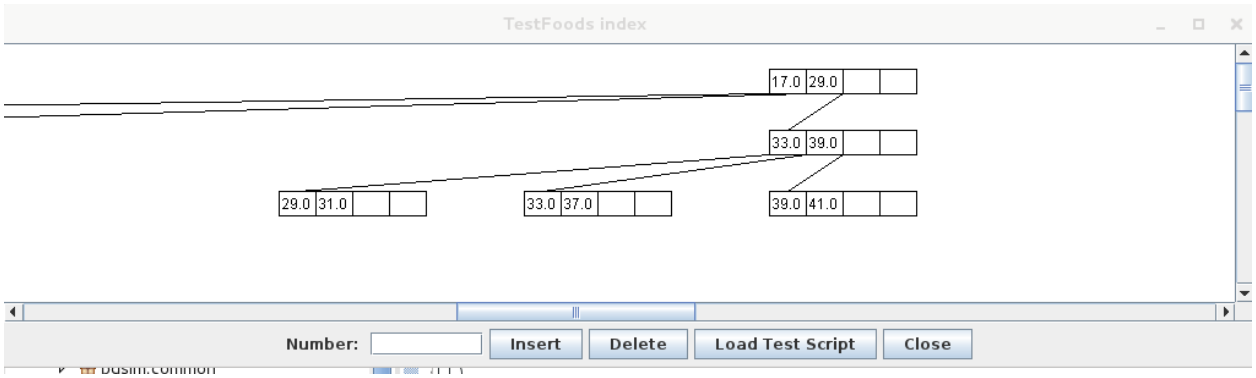
picture2:



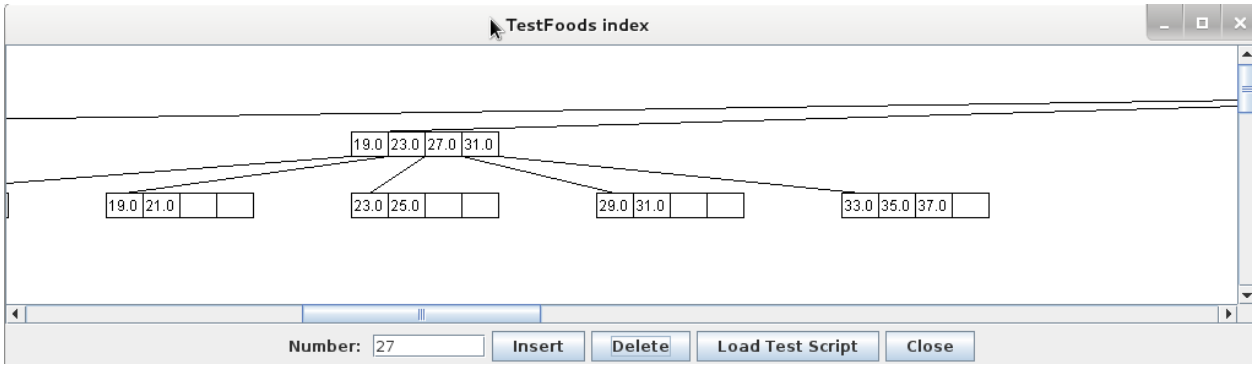
picture3:



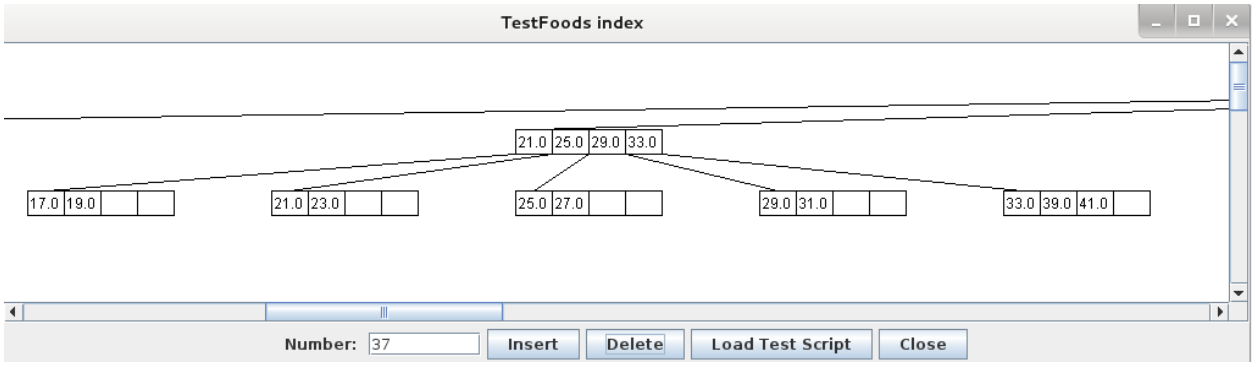
picture4:



from picture1 to picture 3 insert and split the tree.when the stuple is full,the split is not limited on the leafs but also happen to the parents,so we can make sure that insert_in_parent method really works.
rich sibling deletion can be found in picture2 if we delete 27,then we get picture 5:



poor sibling deletion can be found in picture4 in which the leaf and its father all have poor siblings.If we delete 37:



Analysis and tuning:

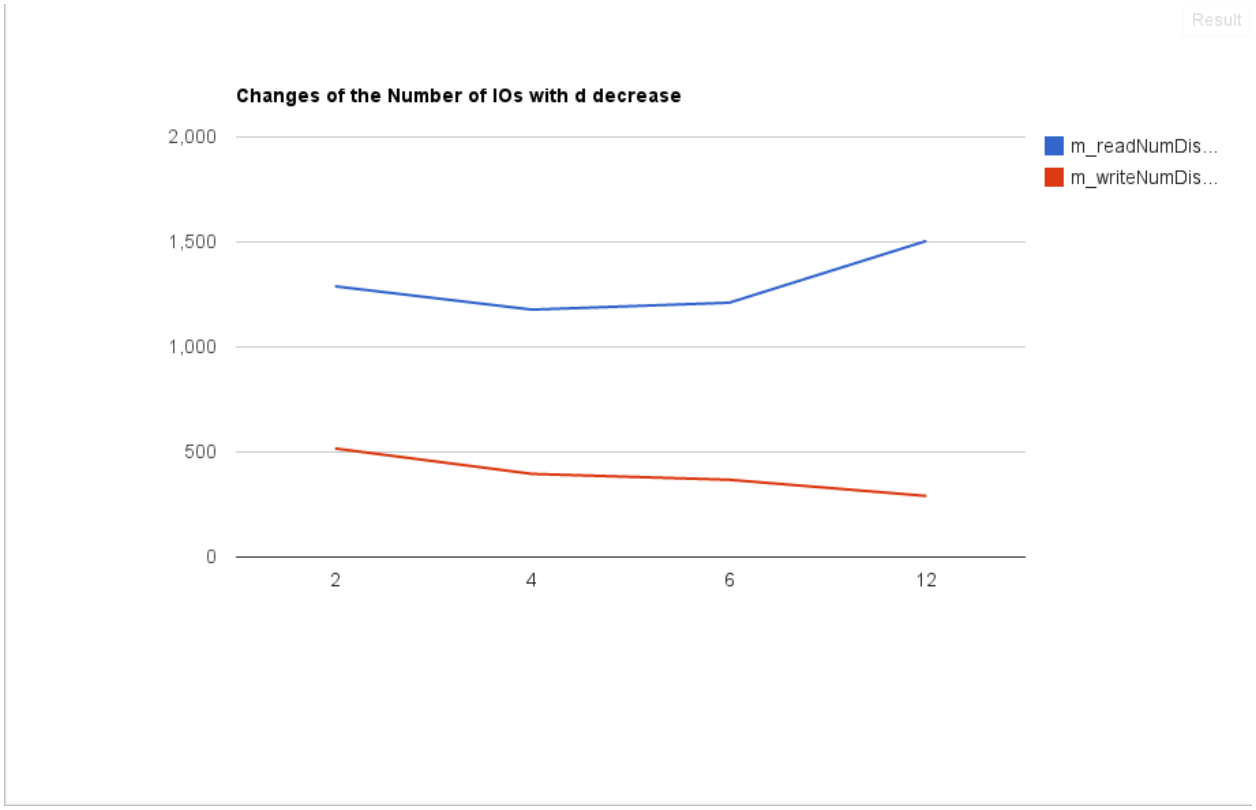
Here we use delete.050.txt as the dataset to test the IO numbers.
when d=2;

[java] m_readNumDiskOps:1289
[java] m_writeNumDiskOps:517

when d=4;
 [java] m_readNumDiskOps:1178
    [java] m_writeNumDiskOps:396

when d=6;
 [java] m_readNumDiskOps:1211
    [java] m_writeNumDiskOps:368

when d=12;
[java] m_readNumDiskOps:1505
    [java] m_writeNumDiskOps:291



as d becomes bigger, m_readNumDiskOps increase, while m_writeNumDiskOps decrease in a relative slow rate. This is quite easy to understand,as d increase,the number of nodes decrease,there are fewer chances to split or merge a node,thus the action of copying decrease,leading to the decrease of m_writeNumDiskOps,while d increase we need to scan more nodes in the node to find the exact key or child we want, this increase with d increase, thus m_readNumDiskOps increase with the scanning action, for d=2,d is so small that the depth has an obvious increase,which has an obvious effect on the increasement of m_readNumDiskOps.