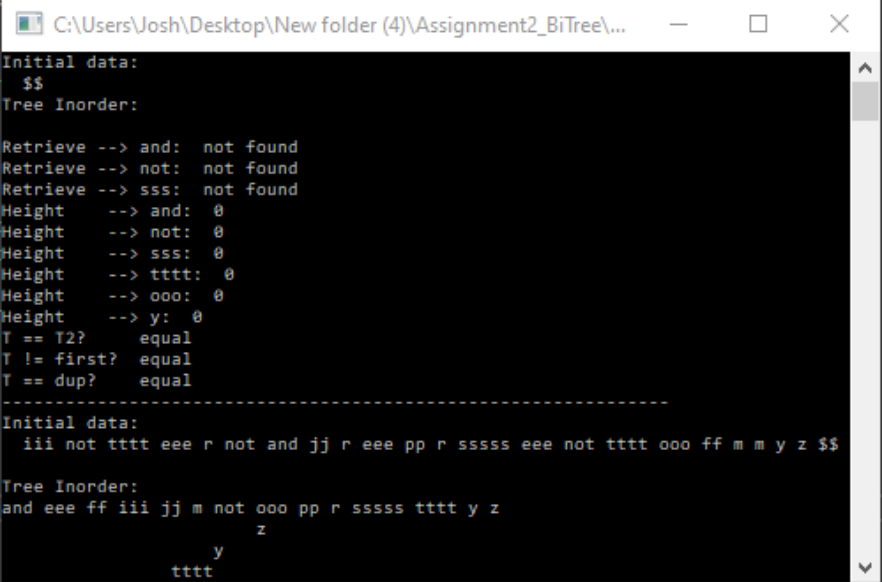
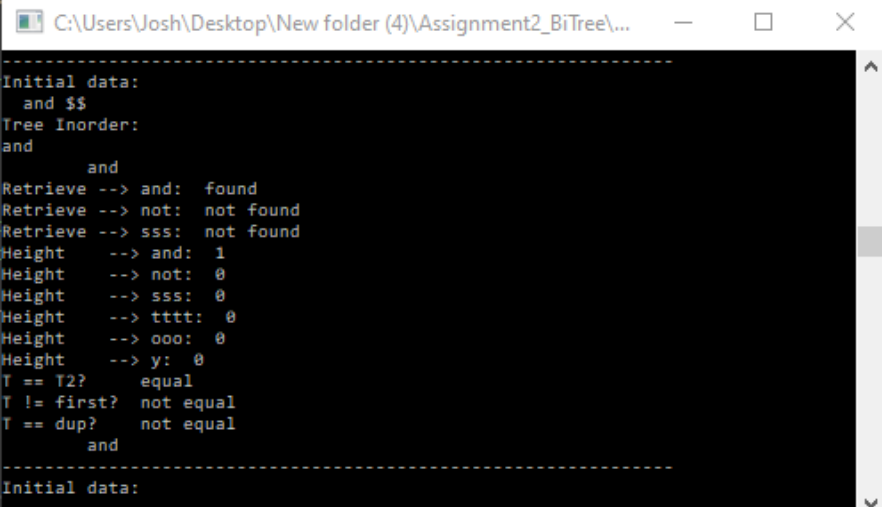
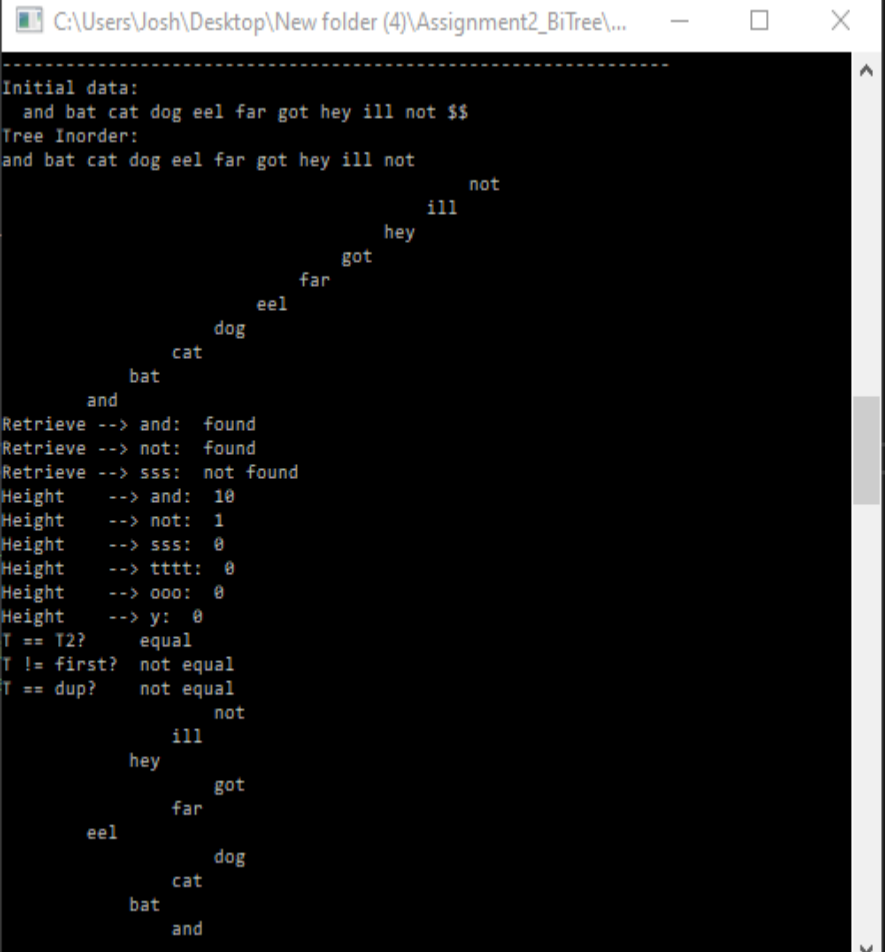
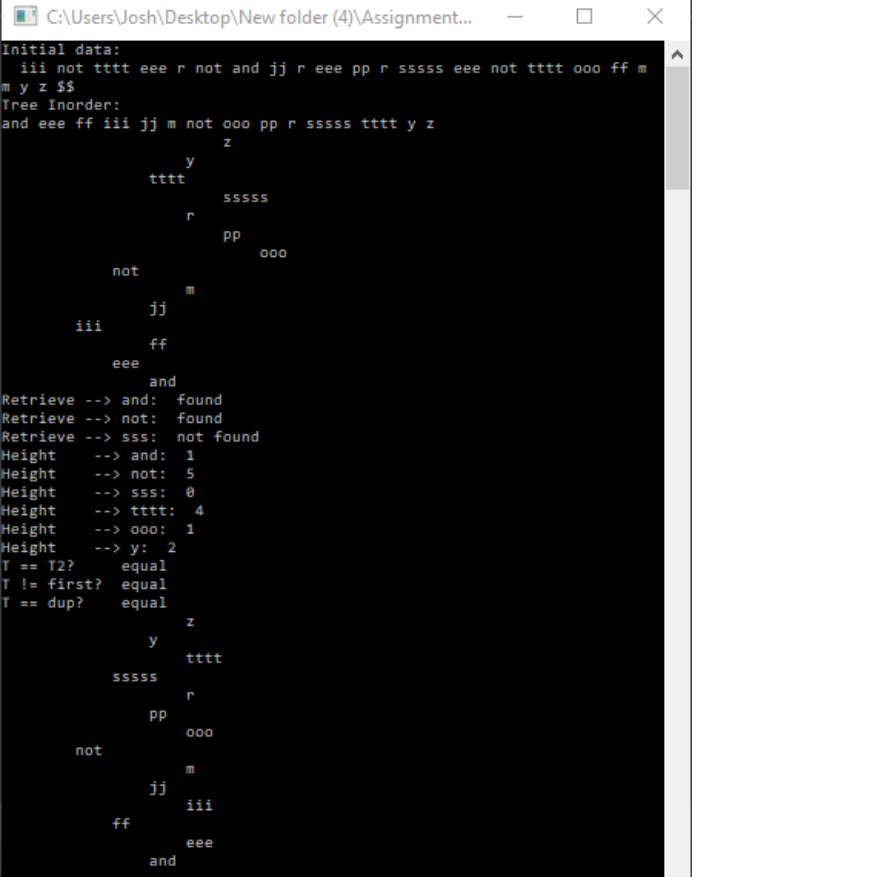


**Table 1: Assignment 2 Test Matrix**

Test Case	Rational	Output + Notes	Picture
Empty Input: \$\$	T should be set to null and the program should run without crashing.	All tests pass, first and dup are set to T, which is null, so null = null. All retrieves should return not found, all heights should be not found (0).	 <pre> C:\Users\Josh\Desktop\New folder (4)\Assignment2_BiTree\... Initial data: \$\$ Tree Inorder:  Retrieve --&gt; and: not found Retrieve --&gt; not: not found Retrieve --&gt; sss: not found Height --&gt; and: 0 Height --&gt; not: 0 Height --&gt; sss: 0 Height --&gt; tttt: 0 Height --&gt; ooo: 0 Height --&gt; y: 0 T == T2? equal T != first? equal T == dup? equal ----- Initial data: iii not tttt eee r not and jj r eee pp r sssss eee not tttt ooo ff m m y z \$\$ Tree Inorder: and eee ff iii jj m not ooo pp r sssss tttt y z z y tttt </pre>
Single tree, root only: and \$\$	Program should not crash because only the root is added.	Value and height are found as usual, everything else matches.	 <pre> C:\Users\Josh\Desktop\New folder (4)\Assignment2_BiTree\... ----- Initial data: and \$\$ Tree Inorder: and and Retrieve --&gt; and: found Retrieve --&gt; not: not found Retrieve --&gt; sss: not found Height --&gt; and: 1 Height --&gt; not: 0 Height --&gt; sss: 0 Height --&gt; tttt: 0 Height --&gt; ooo: 0 Height --&gt; y: 0 T == T2? equal T != first? not equal T == dup? not equal and ----- Initial data: </pre>

<p>Degraded tree, values all greater than the previous: and bat cat dog eel far got hey ill not \$\$</p>	<p>Values should be skewed towards each end of the tree, all methods should work even if it only searches one side.</p>	<p>Retrieval still works as expected although it degrades to linear. Height of the root is correct and the height of the leaf is still 1.</p>	 <p>The screenshot shows a terminal window with the following content:</p> <pre> Initial data:   and bat cat dog eel far got hey ill not \$\$ Tree Inorder: and bat cat dog eel far got hey ill not                                      not                                    ill                                  hey                                got                              far                            eel                          dog                        cat                      bat                    and Retrieve --&gt; and: found Retrieve --&gt; not: found Retrieve --&gt; sss: not found Height --&gt; and: 10 Height --&gt; not: 1 Height --&gt; sss: 0 Height --&gt; tttt: 0 Height --&gt; ooo: 0 Height --&gt; y: 0 T == T2? equal T != first? not equal T == dup? not equal                                      not                                    ill                                  hey                                got                              far                            eel                          dog                        cat                      bat                    and </pre>
<p>Given Values: iii not tttt eee r not and jj r eee pp r sssss eee not tttt ooo ff m m y z \$\$ b a c b a c \$\$ c b a \$\$</p>	<p>The program should match the output which was given to us.</p>	<p>All values match the output and behaves as expected.</p>	 <p>The screenshot shows a terminal window with the following content:</p> <pre> Initial data:   iii not tttt eee r not and jj r eee pp r sssss eee not tttt ooo ff m   m y z \$\$ Tree Inorder: and eee ff iii jj m not ooo pp r sssss tttt y z                                      y                                    z                                  tttt                                sssss                              r                            pp                          ooo                        not                      m                    jj                  iii                ff              eee            and Retrieve --&gt; and: found Retrieve --&gt; not: found Retrieve --&gt; sss: not found Height --&gt; and: 1 Height --&gt; not: 5 Height --&gt; sss: 0 Height --&gt; tttt: 4 Height --&gt; ooo: 1 Height --&gt; y: 2 T == T2? equal T != first? equal T == dup? equal                                      z                                    y                                  tttt                                sssss                              r                            pp                          ooo                        not                      m                    jj                  iii                ff              eee            and </pre>

			<div><div>C:\Users\Josh\Desktop\New folder (4)\Assignment...</div><div><div>Initial data: b a c b a c \$\$ Tree Inorder: a b c</div><div><div>c b a</div><div>Retrieve --&gt; and: not found Retrieve --&gt; not: not found Retrieve --&gt; sss: not found Height --&gt; and: 0 Height --&gt; not: 0 Height --&gt; sss: 0 Height --&gt; tttt: 0 Height --&gt; ooo: 0 Height --&gt; y: 0 T == T2? equal T != first? not equal T == dup? not equal</div><div>c b a</div></div><div><div>Initial data: c b a \$\$ Tree Inorder: a b c</div><div><div>c b a</div><div>Retrieve --&gt; and: not found Retrieve --&gt; not: not found Retrieve --&gt; sss: not found Height --&gt; and: 0 Height --&gt; not: 0 Height --&gt; sss: 0 Height --&gt; tttt: 0 Height --&gt; ooo: 0 Height --&gt; y: 0 T == T2? equal T != first? not equal T == dup? not equal</div><div>c b a</div></div></div></div></div>
--	--	--	--

Figure 1: Valgrind Output

```
joshw488@uw1-320-09: ~  
-----  
Initial data:  
  c b a $$  
Tree Inorder:  
a b c  
      c  
      b  
      a  
Retrieve --> and:  not found  
Retrieve --> not:  not found  
Retrieve --> sss:  not found  
Height   --> and:   0  
Height   --> not:   0  
Height   --> sss:   0  
Height   --> tttt:  0  
Height   --> ooo:   0  
Height   --> y:     0  
T == T2?   equal  
T != first? not equal  
T == dup?  not equal  
      c  
      b  
      a  
-----  
Initial data:  
  
==3396==  
==3396== HEAP SUMMARY:  
==3396==    in use at exit: 0 bytes in 0 blocks  
==3396== total heap usage: 183 allocs, 183 frees, 87,448 bytes allocated  
==3396==  
==3396== All heap blocks were freed -- no leaks are possible  
==3396==  
==3396== For counts of detected and suppressed errors, rerun with: -v  
==3396== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0 from 0)  
joshw488@uw1-320-09:~$
```

Figure 2: Program Running on Linux

```
joshw488@uw1-320-01: ~  
joshw488@uw1-320-01:~$ ./test  
Initial data:  
  iii not tttt eee r not and jj r eee pp r sssss eee not tttt ooo ff m m y z $$  
Tree Inorder:  
and eee ff iii jj m not ooo pp r sssss tttt y z  
      z  
      y  
    tttt  
      sssss  
      r  
      pp  
    ooo  
  not  
    m  
    jj  
  iii  
    ff  
    eee  
  and  
Retrieve --> and: found  
Retrieve --> not: found  
Retrieve --> sss: not found  
Height  --> and: 1  
Height  --> not: 5  
Height  --> sss: 0  
Height  --> tttt: 4  
Height  --> ooo: 1  
Height  --> y: 2  
T == T2? equal  
T != first? equal  
T == dup? equal  
      z  
      y  
    tttt  
      sssss  
      r  
      pp  
    ooo  
  not  
    m  
    jj  
  iii  
    ff  
    eee  
  and  
-----
```

Figure 3: Program Running on Linux Continued

```
joshw488@uw1-320-01: ~  
-----  
Initial data:  
  b a c b a c $$  
Tree Inorder:  
a b c  
      c  
    b  
  a  
Retrieve --> and:  not found  
Retrieve --> not:  not found  
Retrieve --> sss:  not found  
Height  --> and:  0  
Height  --> not:  0  
Height  --> sss:  0  
Height  --> tttt: 0  
Height  --> ooo:  0  
Height  --> y:   0  
T == T2?    equal  
T != first? not equal  
T == dup?   not equal  
      c  
    b  
  a  
-----  
Initial data:  
  c b a $$  
Tree Inorder:  
a b c  
      c  
    b  
  a  
Retrieve --> and:  not found  
Retrieve --> not:  not found  
Retrieve --> sss:  not found  
Height  --> and:  0  
Height  --> not:  0  
Height  --> sss:  0  
Height  --> tttt: 0  
Height  --> ooo:  0  
Height  --> y:   0  
T == T2?    equal  
T != first? not equal  
T == dup?   not equal  
      c  
    b  
  a  
-----  
Initial data:
```