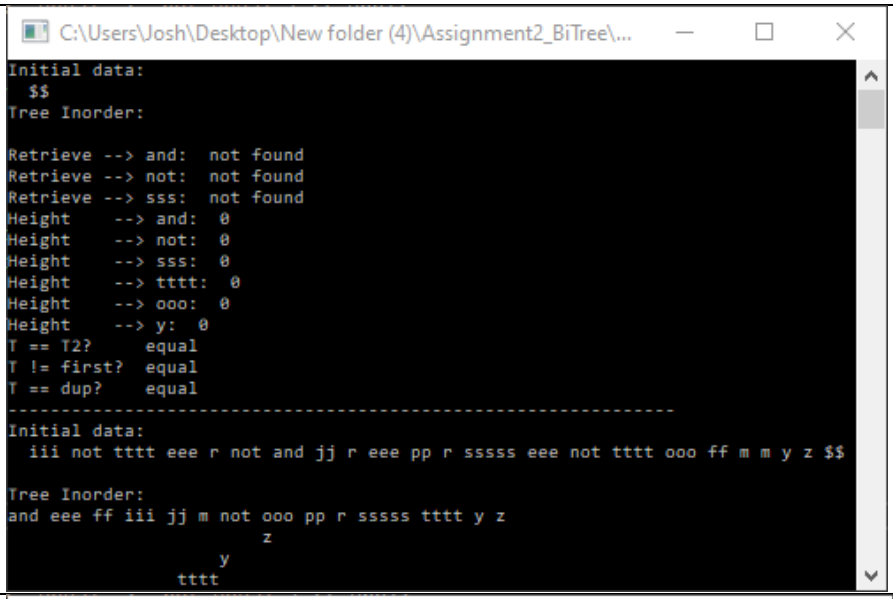
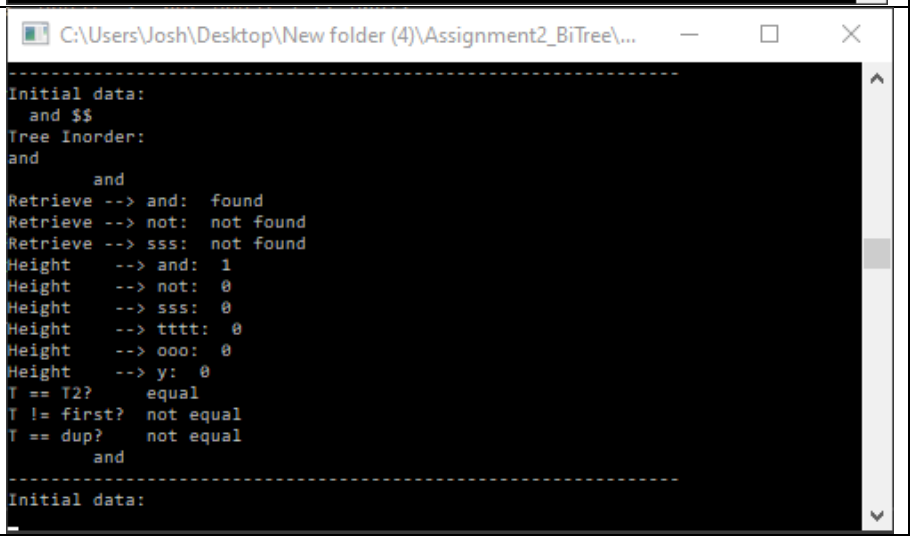
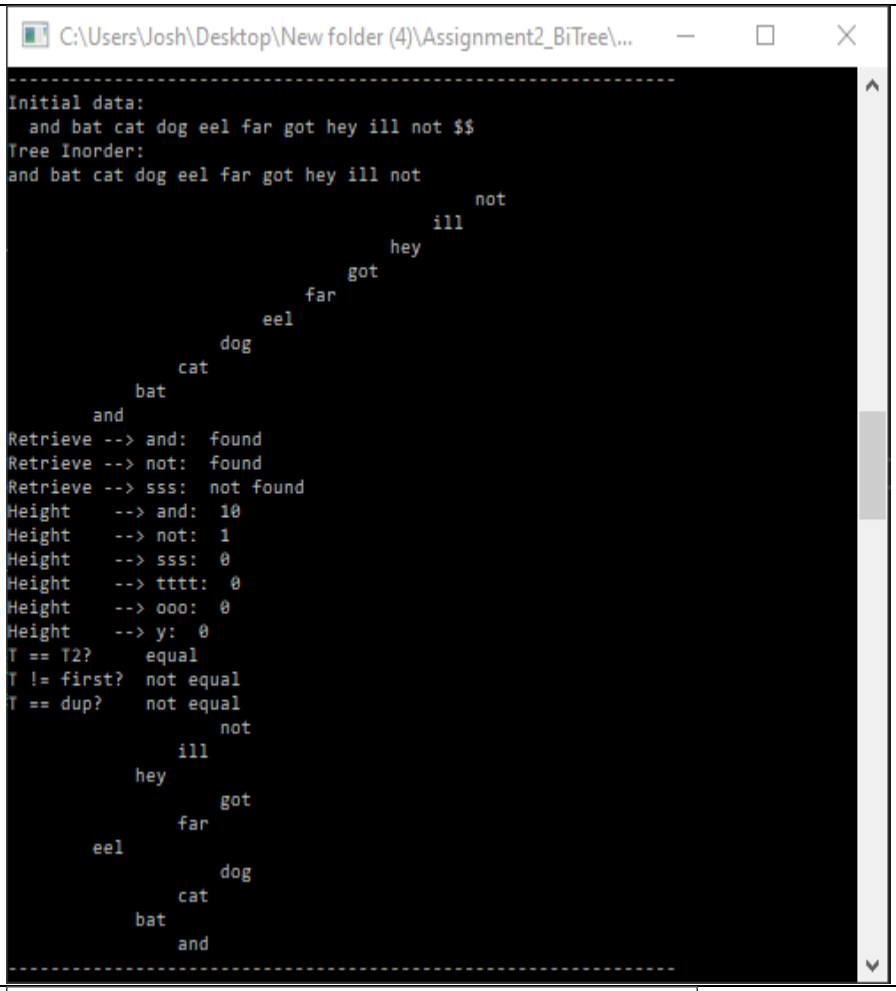
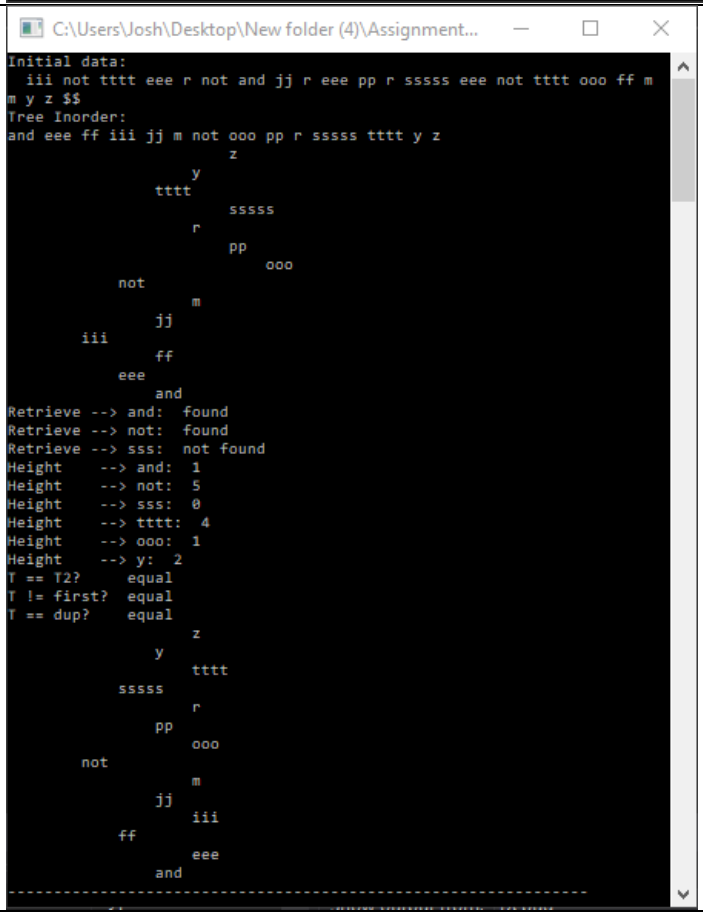
Josh Watts

Table 1: Assignment 2 Test Matrix

Test Case	Rational	Output + Notes	Picture
Empty Input: $$	T should be set to null and the program should run without crashing.	All tests pass, first and dup are set to T, which is null, so null = null. All retrieves should return not found, all heights should be not found (0).	
Single tree, root only: and $$	Program should not crash because only the root is added.	Value and height are found as usual, everything else matches.	

Degraded tree, values all greater than the previous: and bat cat dog eel far got hey ill not $$	Values should be skewed towards each end of the tree, all methods should work even if it only searches one side.	Retrieval still works as expected although it degrades to linear. Height of the root is correct and the height of the leaf is still 1.	
Given Values: iii not tttt eee r not and jj r eee pp r sssss eee not tttt ooo ff m m y z $$ b a c b a c $$ c b a $$	The program should match the output which was given to us.	All values match the output and behaves as expected.	

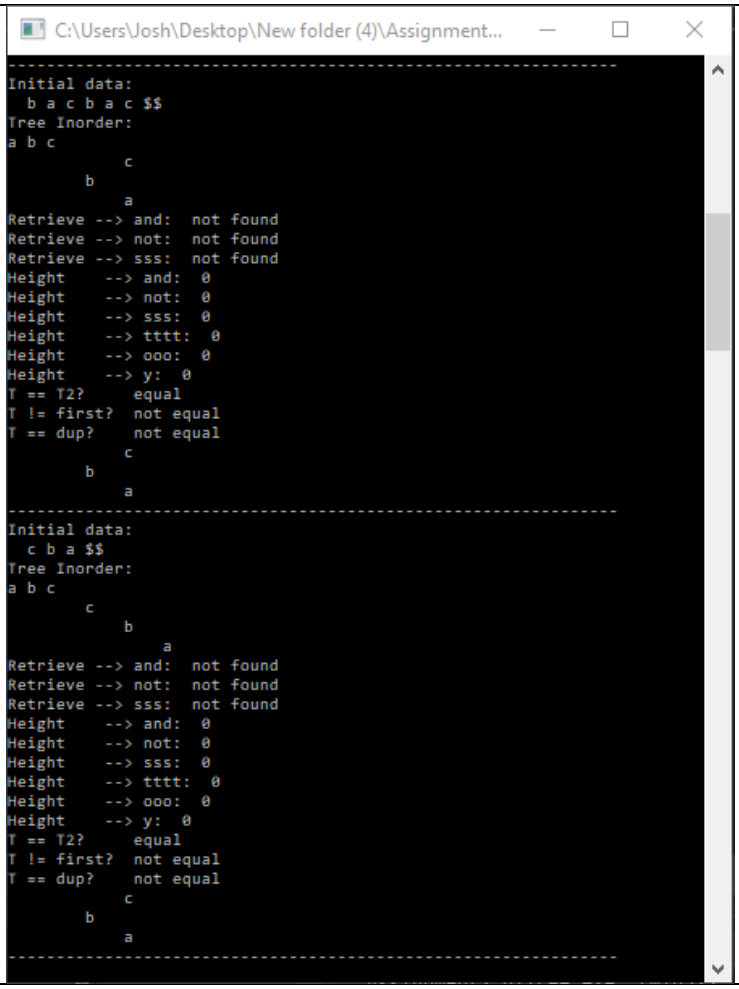

C:\Users\Josh\Desktop\New folder (4)\Assignment...
Initial data:
b a c b a c $$
Tree Inorder:
a b c
c
b
a
Retrieve --> and: not found
Retrieve --> not: not found
Retrieve --> sss: not found
Height --> and: 0
Height --> not: 0
Height --> sss: 0
Height --> tttt: 0
Height --> ooo: 0
Height --> y: 0
T ==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

Figure 1: Valgrind Output

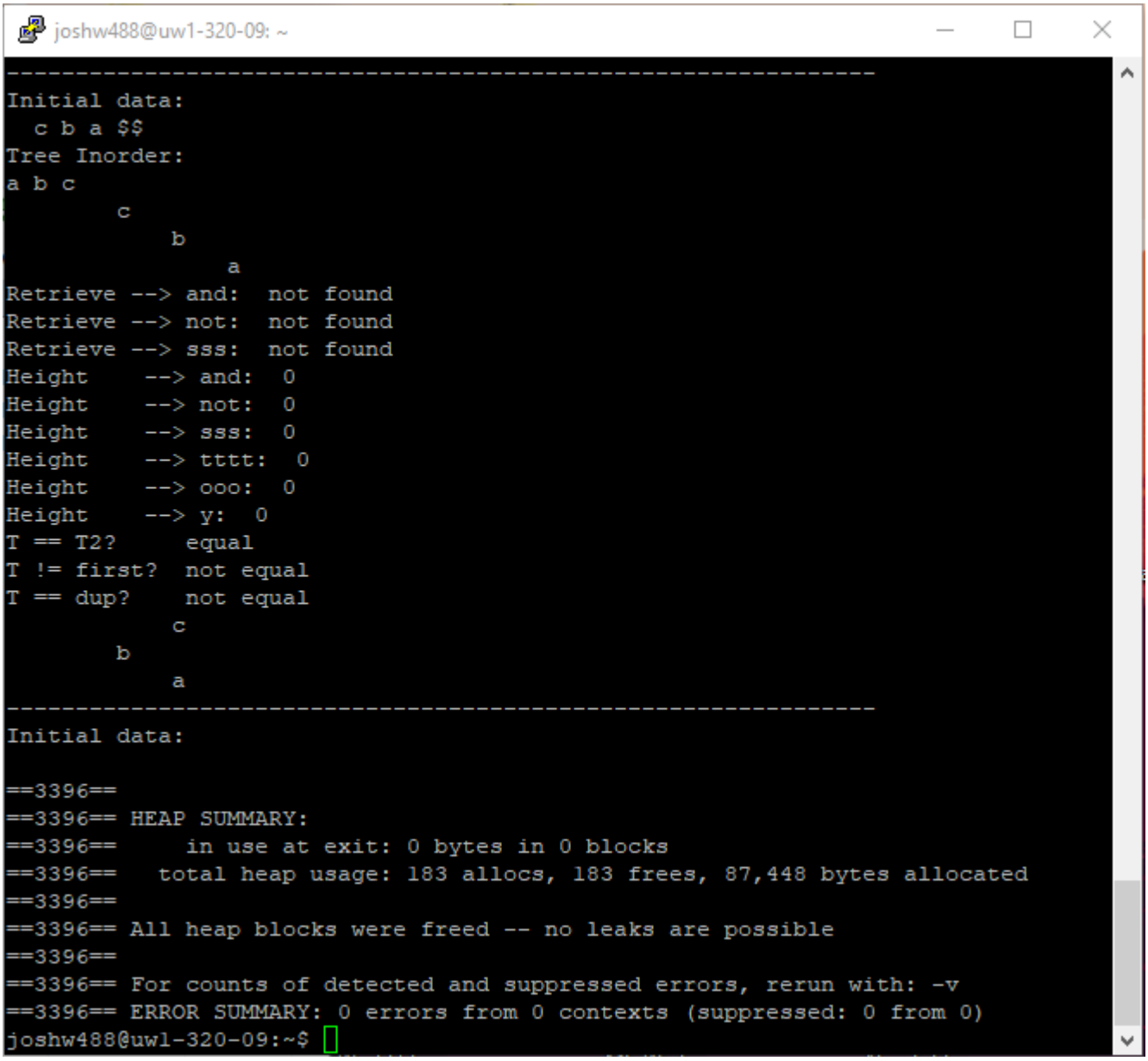


Figure 2: Program Running on Linux

Figure 3: Program Running on Linux Continued