Project 3

William Crutchfield

IntelliJ Ultimate 2017.1.3

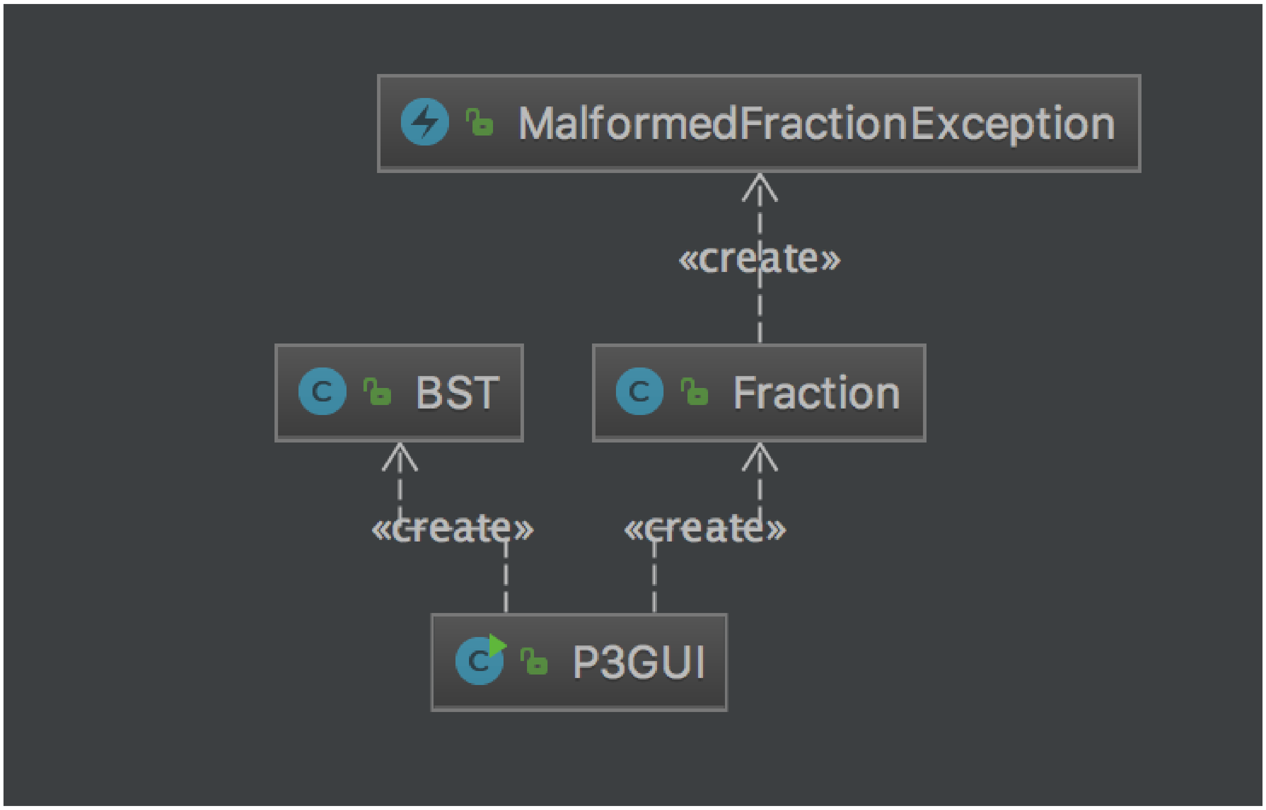
Windows 10

Contents

Assumptions

In this project, I did not make any assumptions. The program will correctly sort fractions, and integers in both ascending and descending order. Also, the program will catch any characters you input. The only exception to this is related to improper fractions. The program will throw an error for a fraction that looks like this “2/3/4” but it will not throw an error for a fraction that looks like this “2/3/”. However, even if a fraction is input in a format such as this “2/3/” it will still output correctly when sorted “2/3”.

UML Diagram

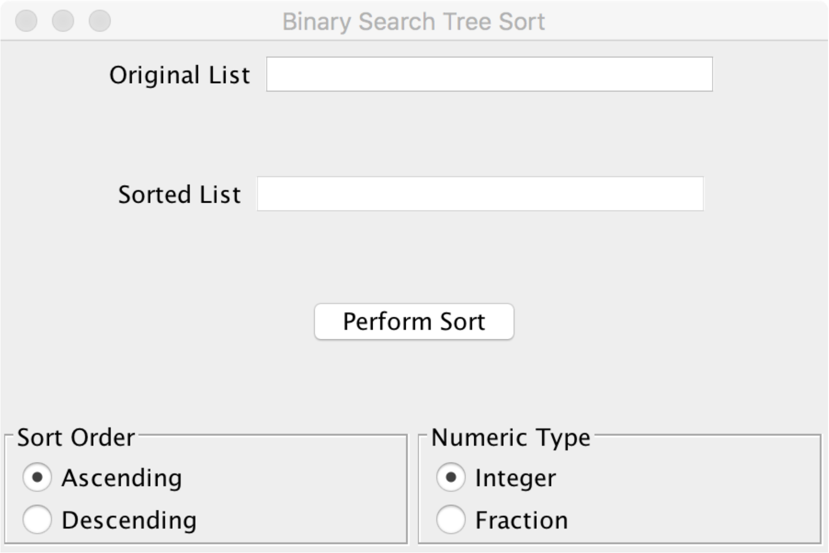

MalformedFractionException
«create»
BST
Fraction
«create»
«create»
P3GUI

Test Cases

Aspect Tested	Input	Expected Output	Actual Output	Test Outcome
Original Integer List (Integers Ascending)	“4 8 2 1 23 16 8 16 3 14 2 10 24”, Ascending	“1 2 2 3 4 8 8 10 14 16 16 23 24”	“1 2 2 3 4 8 8 10 14 16 16 23 24”	Passed
Original Fraction List (Fractions Descending)	“1/2 3/4 3/2 5/8 4/9 7/16 5/32 1/8”, Descending	“3/2 3/4 5/8 1/2 4/9 7/16 5/32 1/8”	“3/2 3/4 5/8 1/2 4/9 7/16 5/32 1/8”	Passed
Integers Descending	“4 63 5 0 -12 5 8”, Descending	“63 8 5 5 4 0 -12”	“63 8 5 5 4 0 -12”	Passed
Fractions Ascending	“1/2 5/32 3/2 4/9”, Ascending	“5/32 4/9 1/2 3/2”	“5/32 4/9 1/2 3/2”	Passed
Malformed Fraction Exception	“1/4 6/3/2 2/5 5/9”, Descending	Error: 6/3/2	Error: 6/3/2	Passed
Malformed Fraction Exception	“1/4 6/3/ 2/5 5/9”, Descending	“6/3 5/9 2/5 1/4”	“6/3 5/9 2/5 1/4”	Passed
Non-Numeric Input (Integers)	“4 8 2 a 23 16 8”	Error: Non-Numeric Input	Error: Non-Numeric	Passed
Non-Numeric Input (Integers)	“4 8 2 1/4 23 16 8”	Error: Non-Numeric Input	Error: Non-Numeric Input	Passed
Malformed Fraction Exception	“1/4 a 6/3 2/5 5/9”	Error: a	Error: a	Passed
Malformed Fraction Exception	“1/4 2 6/3 2/5 5/9”	Error: 2	Error: 2	Passed

Final Product

Upon opening the program, a window will open that looks like this.



Original List

Sorted List

Sort Order

● Ascending

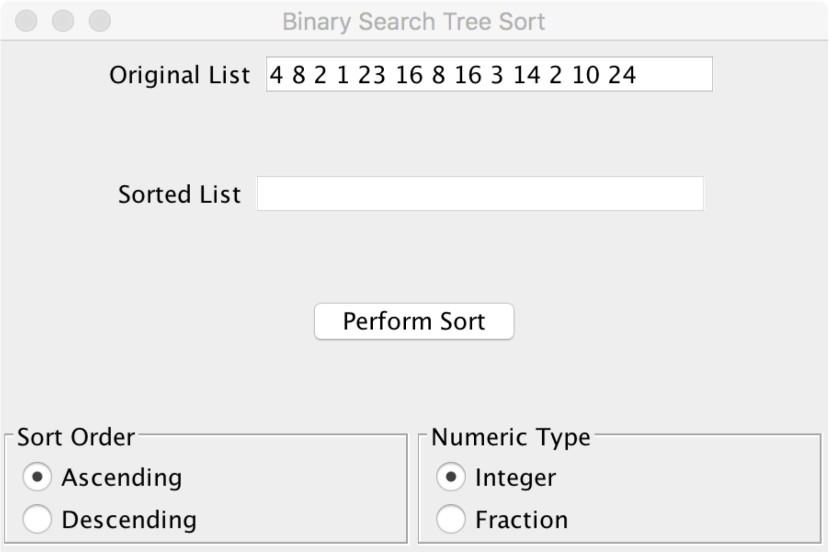
○ Descending

Numeric Type

● Integer

○ Fraction

To start off, we can enter integers into the input text field.



Original List 4 8 2 1 23 16 8 16 3 14 2 10 24

Sorted List

Sort Order

● Ascending

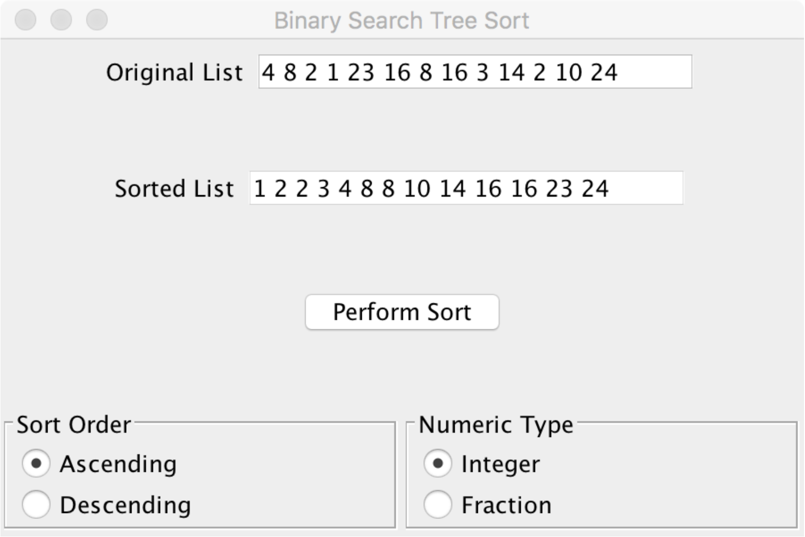
○ Descending

Numeric Type

● Integer

○ Fraction

After entering the integers we can then click “Perform Sort” to sort the integers in ascending order.



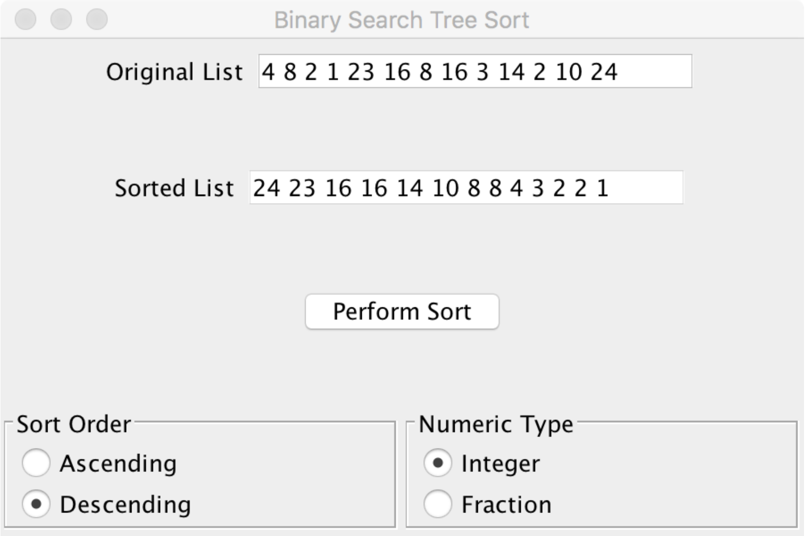
Original List 4 8 2 1 23 16 8 16 3 14 2 10 24

Sorted List 1 2 2 3 4 8 8 10 14 16 16 23 24

Sort Order
● Ascending
○ Descending

Numeric Type
● Integer
○ Fraction

If we would like to sort the integers in descending order, we would just need to click the corresponding radio button and perform the sort again.



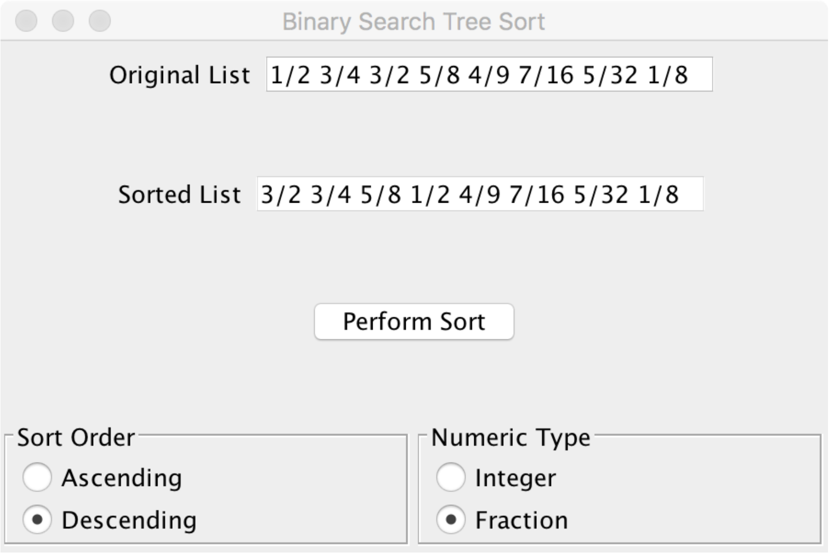
Original List 4 8 2 1 23 16 8 16 3 14 2 10 24

Sorted List 24 23 16 16 14 10 8 8 4 3 2 2 1

Sort Order
○ Ascending
● Descending

Numeric Type
● Integer
○ Fraction

If we would like to sort fractions, we would need to change our input, select the corresponding radio button, and click “Perform Sort” again.



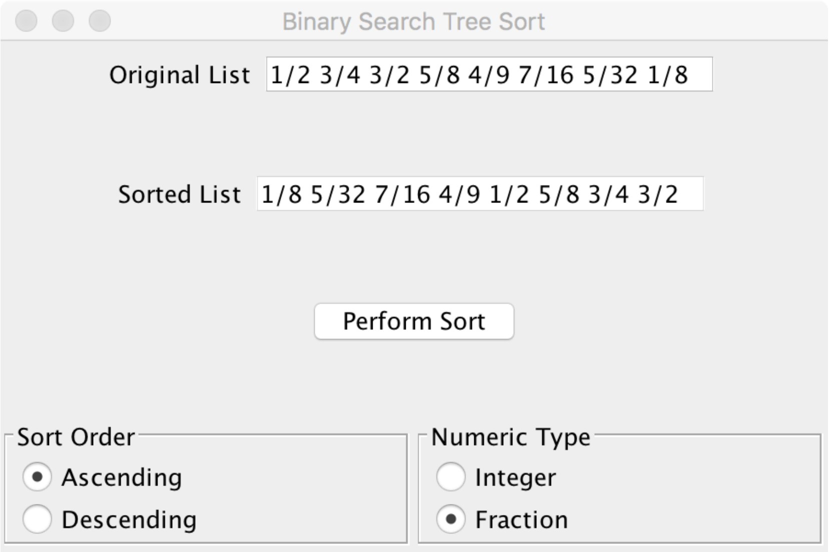
Original List 1/2 3/4 3/2 5/8 4/9 7/16 5/32 1/8

Sorted List 3/2 3/4 5/8 1/2 4/9 7/16 5/32 1/8

Sort Order
○ Ascending
● Descending

Numeric Type
○ Integer
● Fraction

We could then sort our fractions in ascending order, by changing the “Sort Order” and clicking “Perform Sort” once more.



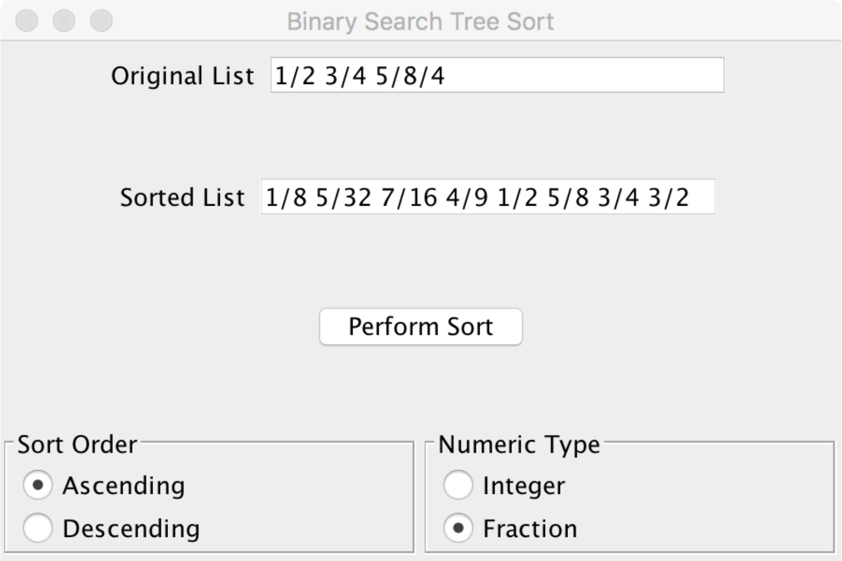
Original List 1/2 3/4 3/2 5/8 4/9 7/16 5/32 1/8

Sorted List 1/8 5/32 7/16 4/9 1/2 5/8 3/4 3/2

Sort Order
● Ascending
○ Descending

Numeric Type
○ Integer
● Fraction

To prove that the program throws an error for a Malformed Fraction we can then change the input and try to perform a sort. I decided to put the improper fraction “5/8/4” in the input. We are then approached with the following.

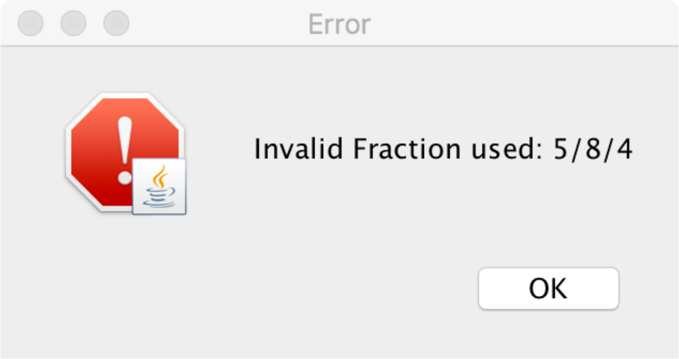

Binary Search Tree Sort

Original List 1/2 3/4 5/8/4

Sorted List 1/8 5/32 7/16 4/9 1/2 5/8 3/4 3/2

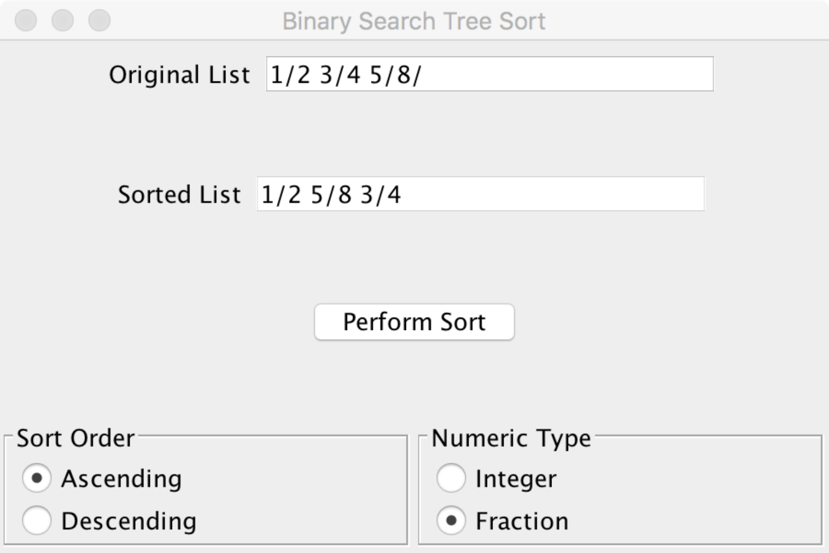
Sort Order
● Ascending
○ Descending

Numeric Type
○ Integer
● Fraction


Error

Invalid Fraction used: 5/8/4

However, if the fraction happened to be “5/8/” this would be the outcome.



Original List 1/2 3/4 5/8/

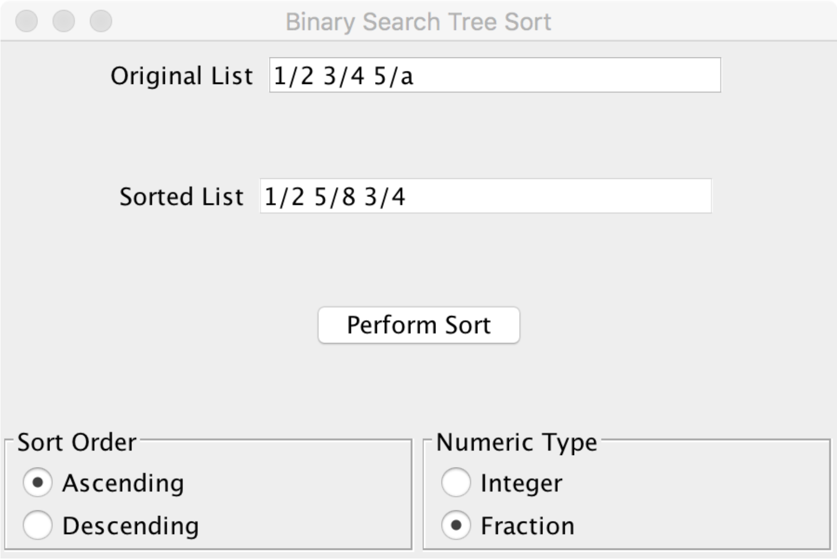
Sorted List 1/2 5/8 3/4

Sort Order
● Ascending
○ Descending

Numeric Type
○ Integer
● Fraction

As you can see, it corrects the fraction to just “5/8”.

Lastly, if we were to enter a Non-Numeric character, when sorting either Fractions or Integers, we would receive an error. To show this, we will now change the fraction “5/8” to “5/a”.



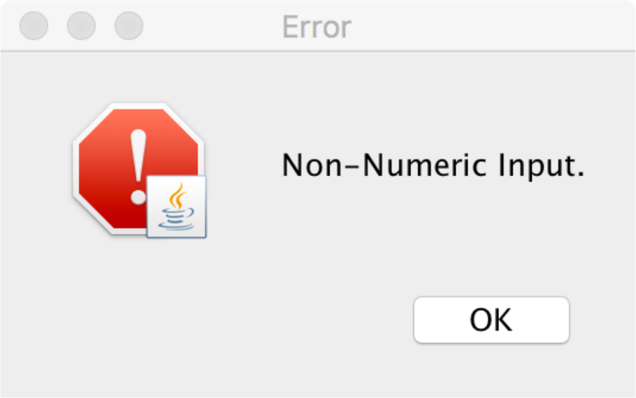
Binary Search Tree Sort

Original List 1/2 3/4 5/a

Sorted List 1/2 5/8 3/4

Sort Order
● Ascending
○ Descending

Numeric Type
○ Integer
● Fraction



Error

Non-Numeric Input.

Lessons Learned

Overall, I thought this project was a great challenge! I was able to get the BST working great when it came to integers, however I had quite a bit of trouble at using the BST on Fractions. After some research, I realized my mistake was something quite simple, but easy to look over due to the fact that I didn't have much experience with the Comparable interface. My main issue was with how I implemented the Comparable interface in my Fractions.java file. I wasn't implementing "Comparable<Fraction>" I was just implementing "Comparable". After realizing this, I was able to effectively create a BST of Nodes containing Fractions. In conclusion, this project was a great way of reaffirming my knowledge of generics, as well as learning how to use the comparable interface effectively with various types of Objects. Please let me know if there's anything I could improve upon!