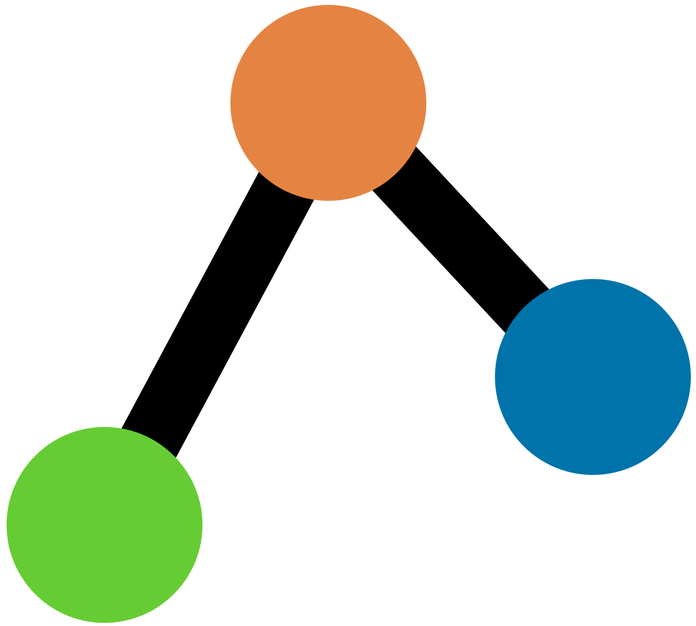


# TreeViz

## Design Analysis

# Introduction

Any new computer engineer or scientist is introduced to data structures and specially tree data structures early on in their lives. At that stage the concepts of different types of tree structures storing data can be a bit hard to grasp; therefore, some visualization techniques emerged to simplify these abstract concepts. These techniques can basically be as simple as a white board and a marker or as complex as a software tool devoted to that purpose.

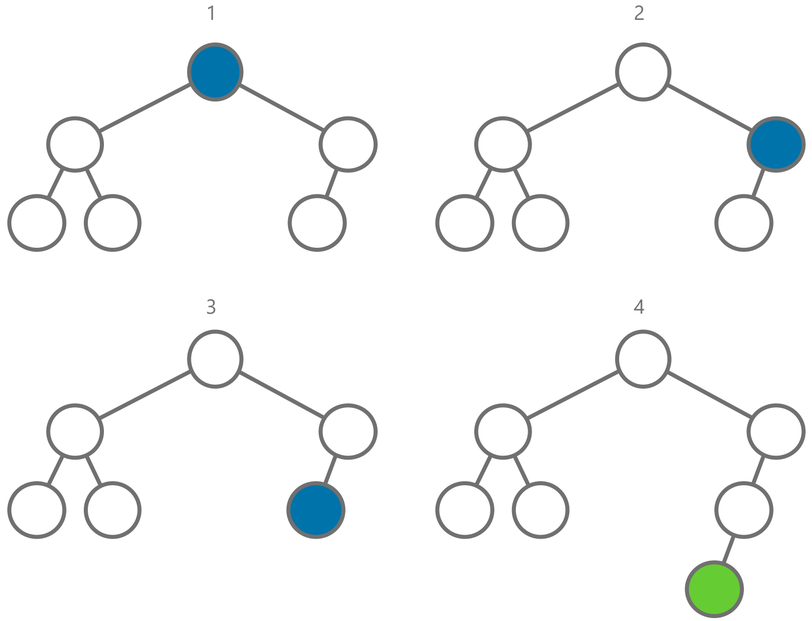
**TreeViz** is an example of a software tool designed to help students better understand the concepts of data structures.

# Student Needs

There are multiple things a students need while trying to understand data structures. These can include the need for more examples, better big picture illustration, comparisons, and easy to use tools.

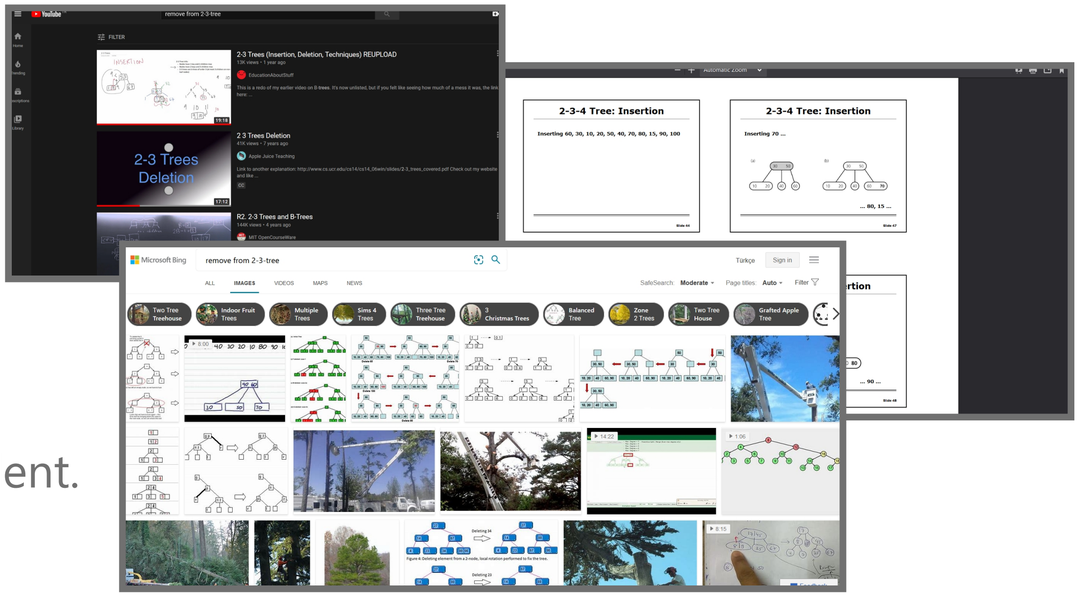
## 1- More examples

Practice is the best way to master any new skill. However, in the case of data structures it is a bit difficult to find resources. In fact there are mainly two ways to find examples:



## Search the internet for solved examples

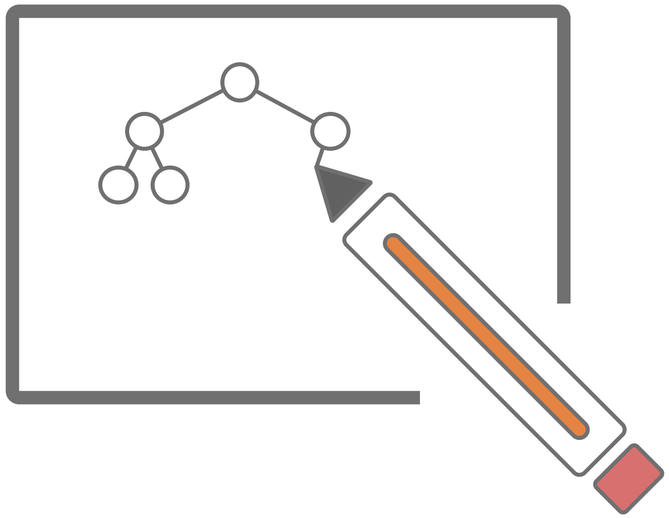
Can be found in images, YouTube videos, or an other universities' professor's slides. The main problems are:




- These are very rare.
- They are not following the same style in constructing the tree, which can be confusing to the student.

## Make one's own examples

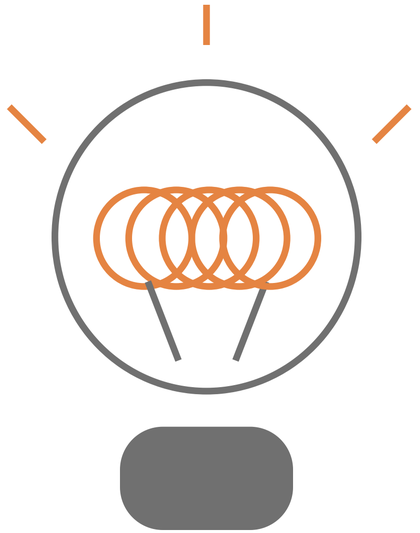
Can be done using any of the existing visualization tools by generating a random example, then inserting or removing from it. There are multiple problems with this approach that are:



- No support for a random tree generation.
- No support for all types of trees.
- No enough options to choose the style of inserting/removing from the tree

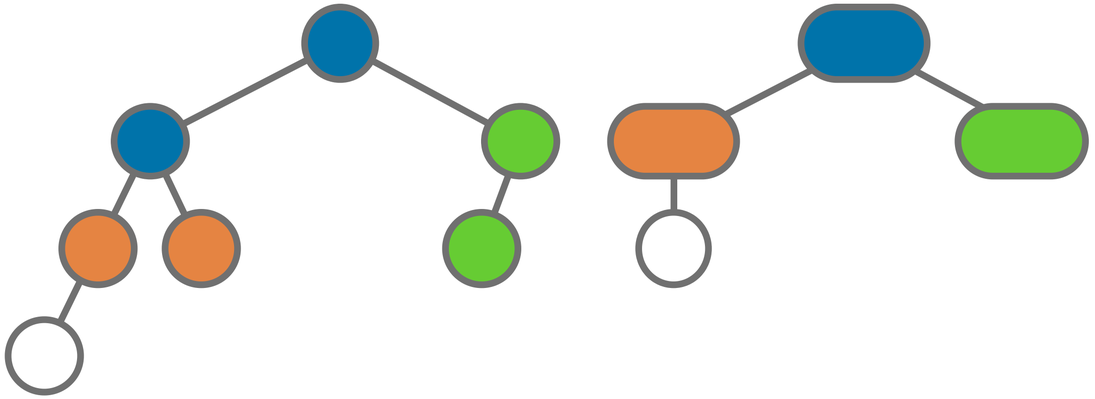
## 2- Big picture illustration

It is usually easier to observe a new subject of study from a high level point of view, before going deep into details. Therefore, it is necessary that a student can observe the difference between tree types on a high level view such that seeing the tree height or speed of building while using a large dataset.



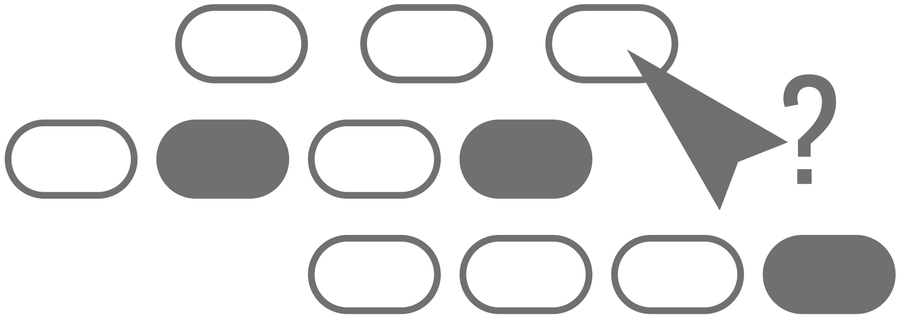
## 3- Trees comparisons

One of the best ways to master a certain subject is by comparing its various methods or comparing it with other subjects. In the case of tree data-structures, it is crucial to compare different tree types and techniques to ensure comprehension



# 4- Easier tools to use

The last thing a student would want to do is spending a long time trying to figure out how to use a certain tool, or if the tool doesn't automate some tasks, such as creating random trees.



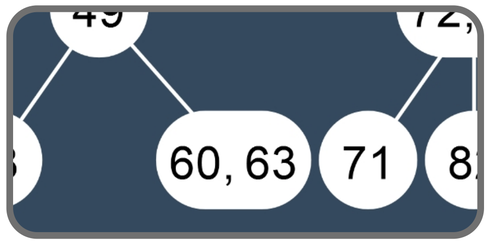
# TreeViz Solutions

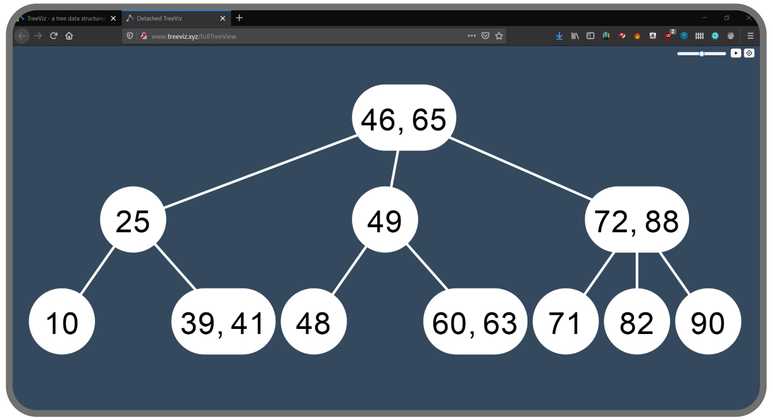
TreeViz has developed solutions in two levels: presentation level and function level.

## Presentation Level Solutions

### 1- Ability to zoom/pan/fullscreen the tree

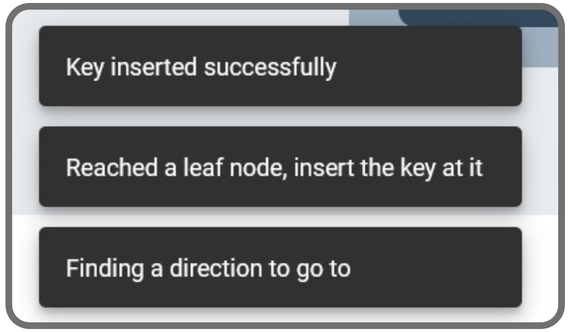
TreeViz provides the ability to zoom to a specific location on the tree, pan to a different location, as well as to make the tree occupy the whole browser window.

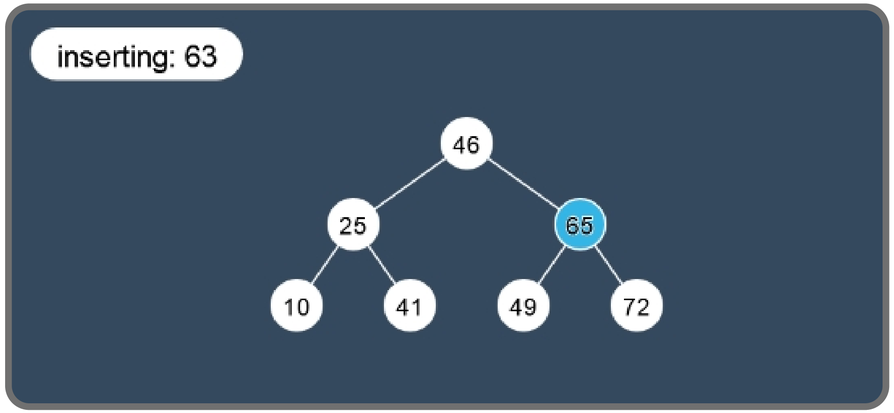


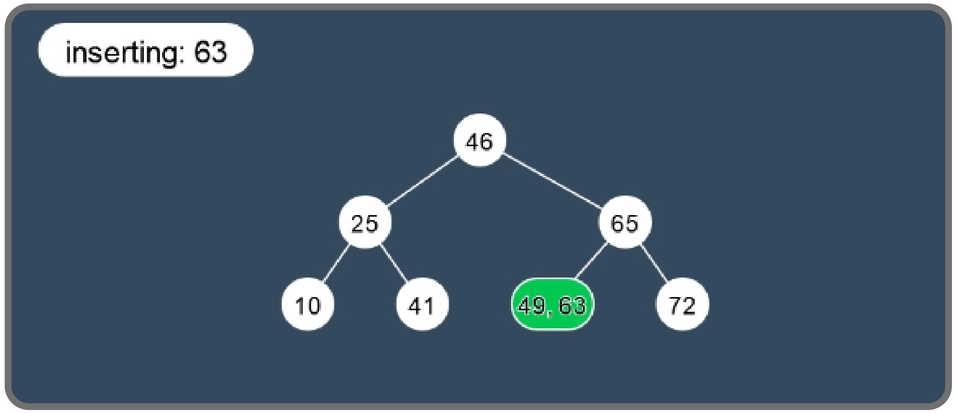



# 2- Tree Animation showing insertion/deletion steps

While constructing a random tree, inserting a value to a tree, or removing a value from a tree an animation would start playing showing the insertion/deletion steps with colors. Instructions would also be shown at the bottom corner in textual format.



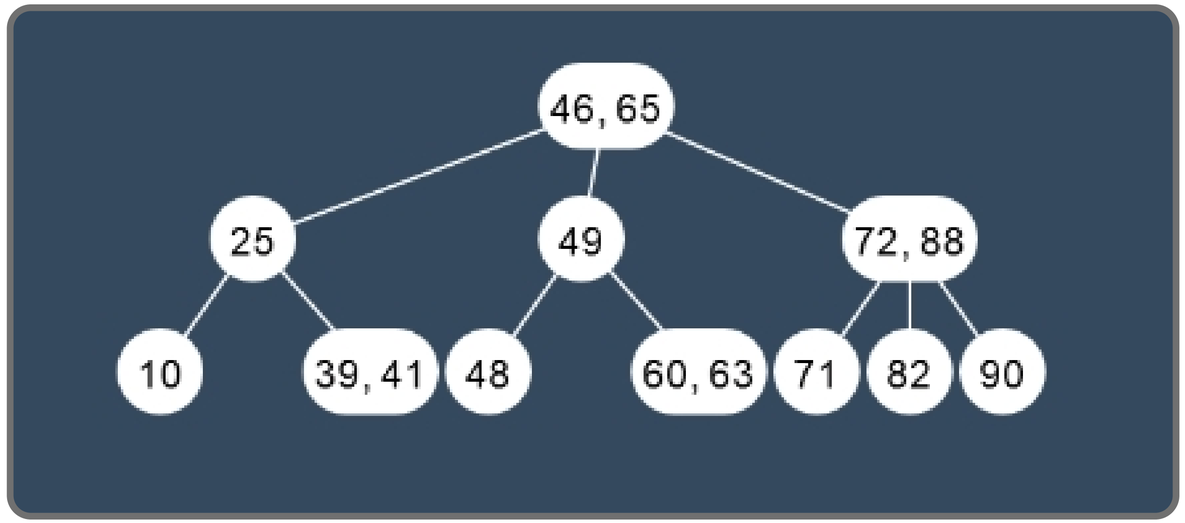



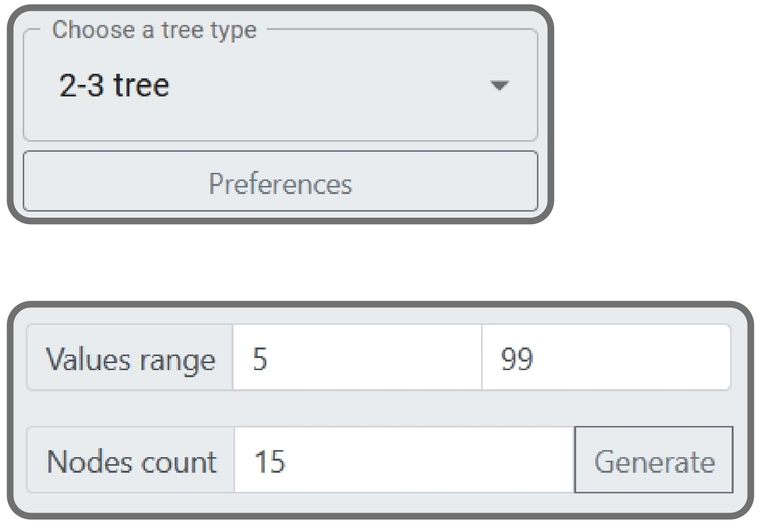



# Function Level Solutions

## 1- Random tree generator

TreeViz lets students generate random trees after specifying the tree type, tree preferences, range of values to have, and the number of nodes wanted. These random trees can serve as practice examples.



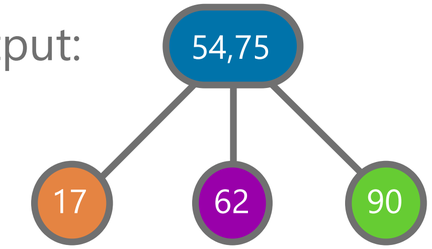



# 2- Tree builder

Sometimes one want to insert a key to a tree printed on paper. It is very difficult to build the tree by inserting numbers since for every insertion sequence a different tree is possible; therefore, TreeViz has developed a treeString structure that is used to describe the tree.

EX: input: **{54,75}({17},{62},{90})**

output:




As we see in the example, the curly braces are used to describe the contents of a node that are numbers separated by commas. The children of a node are encapsulated in parenthesis. such that **{node values}(node's children)**

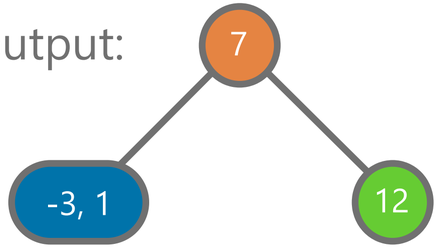
This functionality can be reached using "Describe a tree" input box then "Go"

# 3- Multiple insertions/deletions

For the purpose of saving students time as much as possible, TreeViz has implemented TreeSequence structure. This structure is used to make multiple insertions/deletions at once.

EX: input: **1,4,7,12,d4,-3,-1,d-1**

output:



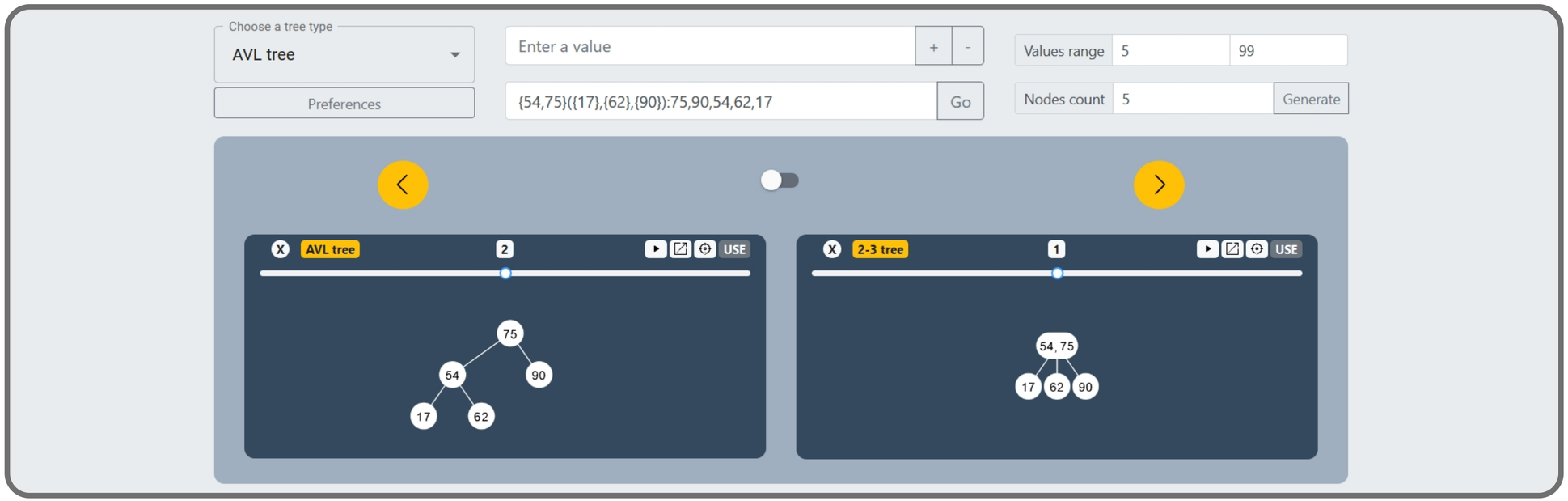

As we see in the example, the integers separated by commas are to be inserted (colored in blue.) on the other hand, the ones that have letter "d" behind them (colored in red) are to be deleted.

This functionality, Just like Tree builder, can be reached using "Describe a tree" input box then "Go"

# 4- Tree conversion

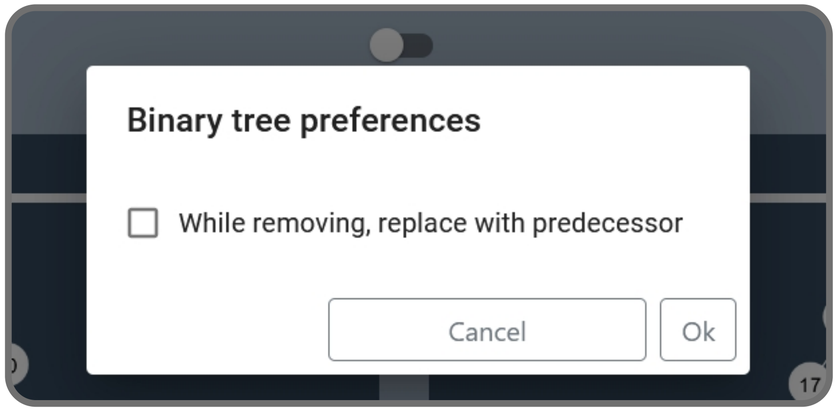
As mentioned earlier, comparison is important in learning; hence, TreeViz has implemented a methods to convert trees from one type to another an compare them.

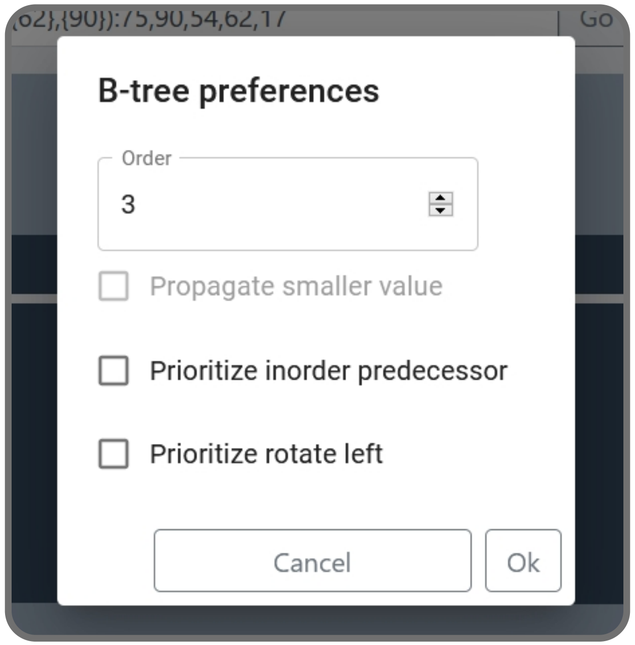
By pressing "USE," the sequence of the tree being viewed is copied to "Describe a tree" input box. Then by choosing a different tree type and pressing "Go," the converted tree would show up to the left of the old one.



# 5- Tree preferences

TreeViz provides the highest number of preferences to cover all possible insertion/deletion techniques. These include, left/right rotation preferences or lower/higher value propagation in the case of B-trees with even order.



# Conclusion

TreeViz is a new tree data-structures visualization tool with an easy to use modern design, as well as a large set of functionality to help students study and master the somewhat complex topic of data-structures.

More information and usage tutorials can be found in TreeViz website at www.treeviz.xyz