CS 1332 Graphics Utilities v1

Georgia Institute of Technology
Spring 2011
David Esposito
desposito6@gatech.edu

We are providing you with a drawing utility which should help with debugging your projects. This will allow you to see a picture of your current data structure rather than looking at traversals.

How To Use It:

1. Download the "CS1332utils.jar" file from T-Square in the resources section.
2. Right click the project in the "Package Explorer" window. Choose "Build Path->Add External Archives…".
3. Browse to the jar file you just downloaded and add it to the project by hitting "Open".
4. Edit your files implement the correct data structure element… For HW02:

```
public class BSTNode<K extends Comparable<K>, V> implements

CS1332utils.Interfaces.BinNode32<Comparable>
```

--- or ---

```
import CS1332utils.Interfaces.BinNode32;

public class BSTNode<K extends Comparable<K>, V> implements BinNode32<K>
```

5. Implementing the methods. Here is the example from HW02 but you can implement them as needed.

```
@Override
public Comparable getContents() {
	return key;
}
```

```
@Override
public void setContents(Comparable arg0) {
	if(arg0==null||arg0.getClass().equals(key.getClass()))
		key = (K) arg0;
}
```

```
@Override
public void setRight(BinNode32<Comparable> arg0) {
	if(arg0 == null || arg0.getClass().equals(this.getClass()))
		right = (BSTNode<K, V>) arg0;
}
```

```
@Override
public void setLeft(BinNode32<Comparable> arg0) {
	if(arg0 == null || arg0.getClass().equals(this.getClass()))
		left = (BSTNode<K, V>) arg0;
}
```

6. Display your current data structure using the Draw class from the utils library (CS1332utils.Display.Draw). For HW02:

```
CS1332utils.Display.Draw.dispTree(root, "myTree.png");
```

--- or ---

```
import CS1332utils.Display.Draw;

Draw.dispTree(root);
```

The methods:

For each data structure there are at least two methods which can be used to display the structure.

- disp__StructureName__(data);
    - Method one will display the structure holding the given data.
- disp__StructureName__(data, filename);
    - Method two takes in the data to construct the structure and also takes in a filename. This file name is used to create an image of the structure on your computer.

Supported Data Structures:

- Lists
    - Linked List
    - Array List
- Heaps
    - Min
    - Max
    - 0 Indexed
    - 1 Indexed
- Hash Table
    - Array Based
    - External Chained
    - 2D Array Based
- Trees
    - Binary Tree

Structures to be Added:

- Lists
    - Circular

- Trees
  - N-ary Trees (MSTs)
- Graph
  - Bi-Directional
  - Directional
  - Adjacency List
  - Connection Matrix

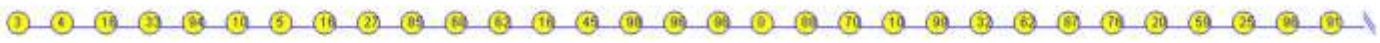
Known Issues with Version 1:

- **Fixed:** The best way of guaranteeing correct output is passing in a deep copy of the data structure. This is because the graphics thread runs after the background code has run. And changes to the data structure after calling the display method will be drawn. Again, passing a deep copy will fix this problem.
- Closing a single window will close all of them. Minimize any unneeded windows rather than closing them.
- Using a filename of an existing file will automatically overwrite it.
- The size of nodes is fixed. Using single or double character data fits best. Using large words will overflow the nodes.

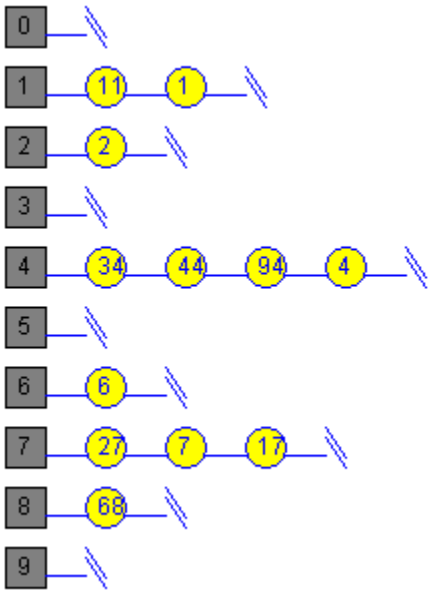
Future Changes:

- Nodes will become variable width fitting all possible toString methods.
- Graphs will be included in future updates.
- **Fixed**: Separate threads will be created, making deep copies of the DS, to allow multiple displays of the same data structure at different points in the program.
- Adding an interface, command line and or GUI, which will allow users to change/save color/size settings.

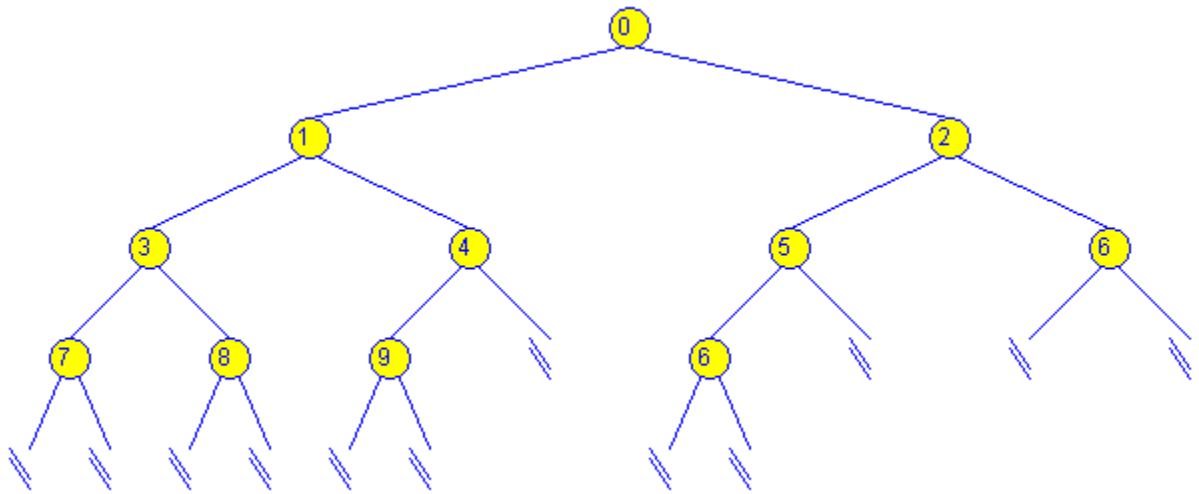
Example Images:



Linked/Array Lists



Hash Tables



Binary Trees and Heaps

For any questions please contact:
David Esposito – desposito6@gatech.edu