**Homework 2**

**Analysis of the random split binary trees**

The code in [1] has a function that creates a random binary tree for a given input treeSize (the number of nodes in the tree). It uses the random split idea at every node [2]. This simple approach does not create a uniform distribution [3]; it is biased and favors (generates more of) balanced trees [2, 4].

Let us test and verify that this program is biased by generating the average number of trees generated for every possible tree height.

For example, if the number of nodes to be created (numNodesCreated) is 7 then there are 429 unique trees. Table 1 shows the distribution of the heights of the trees generated.

Height: Probability

0: 0

1: 0

2: 0.01582

3: 0.4452

4: 0.40604

5: 0.12096

6: 0.01198

Table 1. Probabilities of the heights of 7-node trees generated by the program in [1]

Note that trees with heights of 0 and 1 are not big enough to accommodate 7 nodes. The height cannot be more than 6 in this case and when it is 6 it is degenerate (its shape is like a linked list). When its height is 6 then its shape is like a linked list but it can be zigzagging [5] and thus can have many different unique forms. There are definitely more than just 1.198% of the trees with height 6 but this biased program does not generate them as frequently.

**Your task:** Modify the program to reproduce Table 1. Submit your code and a sample run. If you have difficulty writing a computer program you can submit pseudocode with detailed explanations.

[1] RandomTrees, <https://github.com/olcaykursun/Data-Structures-and-Algorithms>

[2] <https://en.wikipedia.org/wiki/Random_binary_tree#Random_split_trees>

[3] <https://en.wikipedia.org/wiki/Random_binary_tree#Uniformly_random_binary_trees>

[4] <https://stackoverflow.com/questions/56873764/how-to-randomly-generate-a-binary-tree-given-the-node-number>

[5] [https://miro.medium.com/max/1000/0\*fhM65x\_SyLTwqJg7.png](https://miro.medium.com/max/1000/0*fhM65x_SyLTwqJg7.png)

**Understanding Closest Pair**

Study my closest-pair program in [1] and prepare a documentation. It could include one or more of the following.

Explain the code by adding comments and run it and document runtimes etc.,

Give its pseudocode with explanations and/or figures,

Convert it to C++ and compare with my Python implementation,

Improve the code in some way and make a pull request (more details on this later for the interested students).

[1] ClosestPair, <https://github.com/olcaykursun/Data-Structures-and-Algorithms>