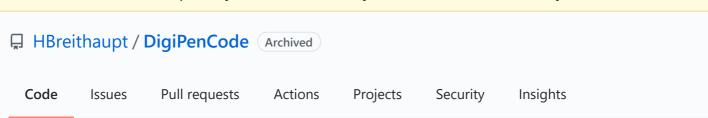
This repository has been archived by the owner. It is now read-only.





DigiPenCode / CS280 / Assignment4 / AVLTree.h

```
HBreithaupt CS280

RX 0 contributors
```

```
\Box
 Raw
      Blame
59 lines (47 sloc) 1.77 KB
    1
 2
    /*!
 3
    \file AVLTree.h
     \author Haven Breithaupt
 4
     \par DP email: h.breithaupt\@digipen.edu
 5
 6
     \par Course: CS280
     \par Assignment 4
 7
     \date
 8
 9
     \brief
 10
     function prototypes for AVLTree
 12
    13
 14
 15
 16
 17
    #ifndef AVLTREE_H
18
 19
    #define AVLTREE_H
    //-----
    #include <stack>
 21
    #include "BSTree.h"
    #include <iostream>
 23
 24
    /// avl tree class
 25
    template <typename T>
 27
    class AVLTree : public BSTree<T>
 28
```

```
29
       public:
30
          /// making typename for compiler to resolve recognizing bintree*
        typedef typename BSTree<T>::BinTreeNode* BinTree;
31
32
          /// defining type for transparency (mostly for myself)
33
34
        typedef std::stack<BinTree> stack;
        AVLTree(ObjectAllocator *OA = 0, bool ShareOA = false);
37
        virtual ~AVLTree();
        virtual void insert(const T& value);
38
        virtual void remove(const T& value);
39
40
41
       private:
42
        // private stuff
43
       void InsertItemAVL(BinTree &tree, const T& value, stack &nodes, int depth);
44
45
       void Balance(stack &nodes, bool Inserting = true);
       void RotateLeft(BinTree &tree);
      void RotateRight(BinTree &tree);
47
      void DeleteItemAVL(BinTree &tree, const T& value, std::stack<BinTree> &nodes);
48
       void AttachRotation(BinTree &parent, BinTree &rotation);
50
       void LeftHeavyBalance(BinTree &y, std::stack<BinTree> &nodes);
      void RightHeavyBalance(BinTree &y, std::stack<BinTree>& nodes);
51
       unsigned int count nodes(BinTree &Root);
53
54
     };
56
    #include "AVLTree.cpp"
57
    #endif
59
     //-----
```