云南大学数学与统计学院
上机实践报告

课程名称：数据结构与算法实验	**年级：**2015 级	**上机实践成绩：**
指导教师：陆正福	**姓名：**刘鹏	
上机实践名称：树结构实验	**学号：**20151910042	**上机实践日期：**2017-05-03
上机实践编号：No.08	**组号：**	**上机实践时间：**上午三、四节

一、实验目的

1. 熟悉与树和二叉树有关的数据结构与算法；
2. 熟悉主讲教材 Chapter 8 的代码片段。

二、实验内容

三、实验平台

Windows 10 Enterprise 中文版；
Python 3.6.0；
Wing IDE Professional 6.0.2-1 集成开发环境。

四、实验记录与实验结果分析

```
# 8.4.6 Euler Tours and the Template Method Pattern

class EulerTour:
    """Abstract base class for performing Euler tour of a tree.

    _hook_previsit
    """
    def __init__(self,tree):
        """Prepare an Euler tour template for given tree."""
        self._tree = tree

    def tree(self):
        """Return reference to the tree being traversed."""
        return self._tree

    def execute(self):
        """Perform the tour and return any result from post visit of root."""
        if len(self._tree) > 0:
            return self._tour(self._tree.root(),0,[])   # start the recursion

    def _tour(self,p,d,path):
        """Perform tour of subtree rooted at Position p.

        p      Position of current node being visited
        d      depth of p in the tree
        path    list of indices of children on path from root to p
        """
        self._hook_previsit(p,d,path)                # "pre visit" p
        results = []
        path.append(0)         # add new index to end of path brfore recursion
        for c in self._tree.children(p):
          results.append(self._tour(c,d+1,path))  # recur on child's subtree
          path[-1] += 1      # increment index
      path.pop()            # remove extraneous index from end of path
      answer = self._hook_postvisit(p,d,path,results) # "post visit" p
      return answer

   def _hook_previsit(sellf,p,d,path):              # can be overriden
      pass

   def _hook_postvisit(self,p,d,path,results):        # can be overriden
      pass

class PreorderPrintIndent(EulerTour):
   def _hook_previsit(self,p,d,path):
      print(2 * d * ' ' + str(p.element()))

class PreorderPrintIndentedLaveledTour(EulerTour):
   def _hook_previsit(self,p,d,path):
      label = '.'.join(str(j+1) for j in path)    # labels are one-indexed
      print(2 * d * ' ' + label,p.element())

class ParenthesizeTour(EulerTour):
   def _hook_previsit(self,p,d,path):
      if path and path[-1] > 0:                # p follows a sibling
          print(', ',end=())                  # so preface with comma
      print(p.element(),end='')                # then print element
      if not self.tree().is_leaf(p):            # if p has children
          print('（',end=())                   #print opening parenthesis

   def _hook_postvisit(self,p,d,path,results):
      if not self.tree().is_leaf(p):            # if p has children
          print('）',end=())                   # print closing parenthesis

class DiskSpaceTour(EulerTour):
   def _hook_postvisit(self,p,d,path,results):
      # we simply add space associated with p to that of its subtrees
      return p.element().space + sum(results)

# The Eulewr Tour Traversal of a Binary Tree
class BianryEulerTour(EulerTour):
   """Abstrct base class for performing Euler Tour of a binary tree.

   This version includes an additional _hook_invisit that is called after the
   tour of the left subtree (if any), yet before the tour of the right subtree
   (if any).

   Note: Right child is always assigned index 1 in path, even if no left
   sibling.
  """
```

```

#------------------------------ my main function ------------------------------
```

五、实验体会

Translation：

Chapter 8 Trees

*第八章 树

8.1 General Trees

*8.1 节 一般的树

Productivity experts say that breakthroughs come by thinking “nonlinearly.” In this chapter, we discuss one of the most important nonlinear data structures in computing – ***trees***. Tree structures are indeed a breakthrough in data organization, for they allow us to implement a host of algorithms much faster than when using linear data structures, such as array-based lists or linked lists. Trees also provide a natural organization for data, and consequently have become ubiquitous structures in file systems, graphical user interfaces, databases, Web sites, and other computer systems.

*

END

六、参考文献

[1] Michael T. Goodrich, Roberto Tamassia, Michael H. Goldwasser, *Data Structures and Algorithms in Python*, Chapter 4

[2] 实验教材：汪萍，陆正福等编著 数据结构与算法的问题与实验 第 1 章