Southwestern University of Finance and Economics
Course Syllabus

西南财经大学教学实施方案

Course Information
课程信息

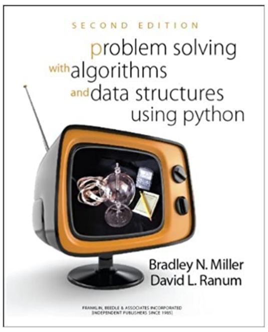
Course Name 课程名	Course Code 课程代码	Credit 学分	Class Hours 学时
Data Structures 数据结构（英）	CST906	3	51 (27 theory+24 practice)

Instructor
任课教师简介

Dr. Guo Jingcai, Ph.D., Associate Professor

School of Computing and Artificial Intelligence, SWUFE.
Block B, Tongbo Building, Liulin Campus, Chengdu.
jcguo@swufe.edu.cn

Textbook and Resources
教材介绍

	Textbook **教材**	**Supplementary Reading** **课外阅读**
Title 题目	 SECOND EDITION problem solving with algorithms and data structures using python Bradley N. Miller David L. Ranum Problem Solving with Algorithms and Data Structures Using Python (2nd edition)	Copies of supplementary texts will be provided in class when necessary. 必要时将在课堂上提供补充文本的复印件。
Author 作者	Bradley N. Miller, David L. Ranum	
Publisher 出版社	Franklin, Beedle & Associates	
Publish time 出版时间	August 22, 2011	

Course Overview
课程简介

An efficient algorithm is usually supported by a set of well-organized data structures that allow the algorithm to run and manipulate the data efficiently. In this course, we focus on the common data structures that are used in various computational problems. You will learn how these data structures are implemented in different programming languages and will practice implementing them in our programming assignments. Topics include the principle of object-oriented programming, basic data structures, e.g., stacks, queues, lists, hash tables, trees, graphs, etc., searching and sorting algorithms, the basics of algorithmic analysis, etc. The concepts and principle of data structures and some related algorithms will also be practiced in projects.

一个高效的算法通常基于一组高效组织的数据结构，以让算法能够高效地运行并对数据进行操作。在本课程中，我们将讨论在各种计算问题中使用的基本数据结构。您将学习如何在编程语言中实现这些数据结构，并将在我们的编程任务中练习如何实现它们。本课程主要内容包括面向对象编程的基本原理、基本数据结构（包括堆栈、队列、列表、哈希表、树和图）、搜索和排序算法、算法分析的基础知识。数据结构、算法的概念和原理同样也将在与列表，二叉树，哈希表，排序和图等相关的工程中得到实践。

Course Objective and Learning Outcomes
课程目标与学习成果

Understand and master the concept of linear table, stack, queue, binary tree, tree, graph and other basic data structures; Understand and master the algorithm flow of basic operations such as adding, modifying, deleting and searching elements in various data structures; Understand and master the basic properties of binary tree, binary search tree, highly balanced binary tree, heap, tree and other nonlinear data structures. Understand and master two common linear table retrieval methods: sequential search and binary search. Understand and master insertion sort, selection sort, bubble sort, quicksort, merge sort, heap sort and other common sorting methods; Understand and master commonly used sorting algorithm; Understand and master the basic concept of the graph, the storage methods of the graph and the implementation of them, understand the graph traversal, shortest path, minimum spanning tree and other common algorithms.

理解并掌握线性表、栈、队列、二叉树、树、图等基本数据结构的概念；理解并掌握各类数据结构中元素的增加、修改、删除、查找等基本操作的算法流程；理解并掌握二叉树、二叉检索树、高度平衡二叉树、堆、树等非线性数据结构的基本性质；理解并掌握顺序查找、二分查找两种常见的线性表检索方法；理解并掌握插入排序、选择排序、冒泡排序、快速排序、归并排序、堆排序等常见的排序方法；理解并掌握常见的外排序算法；理解并掌握图的基本概念、图的存储方法和实现方法，理解图的遍历、最短路径、最小生成树等方面的常用算法。

Requirements and/or Prerequisites (if available)
课程要求或先决条件

Python Language Programming, Discrete Mathematics, etc.
Python 语言编程、离散数学等。

Note: Since the basic data structures are commonly used in various programming languages, it is thus not critical to limit this course to Python if you are familiar with other languages. Nevertheless, some basic knowledge will also be introduced in this course.

Contents 教学内容

No. 序号	Topic 主题	Main knowledge 主要知识点	Hours 课时
1.	Introduction of Data Structures and Algorithms 数据结构与算法介绍	* Objectives * Getting Started * What Is Computer Science? * What Is Programming? * Why Study Data Structures and Abstract Data Types? * Why Study Algorithms? * Review of Basic Python * Getting Started with Data * Input and Output * Control Structures * Exception Handling * Defining Functions * Object-Oriented Programming in Python: Defining Classes * Summary	**4**
2.	Algorithm Analysis 算法分析	* Objectives * What Is Algorithm Analysis? * Big-O Notation * An Anagram Detection Example * Performance of Python Data Structures * Lists * Dictionaries * Summary	**2**
3.	Basic Data Structures 基本数据结构	* Objectives * What Are Linear Structures? * What is a Stack? * The Stack Abstract Data Type * Implementing a Stack in Python * Simple Balanced Parentheses * Balanced Symbols (A General Case) * Converting Decimal Numbers to Binary Numbers * Infix, Prefix and Postfix Expressions * What Is a Queue? * The Queue Abstract Data Type	**4**

		* Implementing a Queue in Python * Simulation: Hot Potato * Simulation: Printing Tasks * What Is a Deque? * The Deque Abstract Data Type * Implementing a Deque in Python * Palindrome-Checker * Lists * The Unordered List Abstract Data Type * Implementing an Unordered List: Linked Lists * The Ordered List Abstract Data Type * Implementing an Ordered List * Summary	
4.	Recursion 递归	* Objectives * What Is Recursion? * Calculating the Sum of a List of Numbers * The Three Laws of Recursion * Converting an Integer to a String in Any Base * Stack Frames: Implementing Recursion * Introduction: Visualizing Recursion * Sierpinski Triangle * Complex Recursive Problems * Tower of Hanoi * Exploring a Maze * Dynamic Programming * Summary	**2**
5.	Sorting and Searching 排序和查找	* Objectives * Searching * The Sequential Search * The Binary Search * Hashing * Sorting * The Bubble Sort * The Selection Sort * The Insertion Sort * The Shell Sort * The Merge Sort * The Quick Sort * Summary	**3**
6.	Trees and Tree Algorithms 树和树算法	* Objectives * Examples of Trees * Vocabulary and Definitions * List of Lists Representation * Nodes and References * Parse Tree * Tree Traversals * Priority Queues with Binary Heaps * Binary Heap Operations * Binary Heap Implementation * Binary Search Trees * Search Tree Operations	**6**

		* Search Tree Implementation * Search Tree Analysis * Balanced Binary Search Trees * AVL Tree Performance * AVL Tree Implementation * Summary of Map ADT Implementations * Summary	
7.	Graphs and Graph Algorithms 图和图算法	* Objectives * Vocabulary and Definitions * The Graph Abstract Data Type * An Adjacency Matrix * An Adjacency List * Implementation * The Word Ladder Problem * Building the Word Ladder Graph * Implementing Breadth First Search * Breadth First Search Analysis * The Knight's Tour Problem * Building the Knight's Tour Graph * Implementing Knight's Tour * Knight's Tour Analysis * General Depth First Search * Depth First Search Analysis * Topological Sorting * Strongly Connected Components * Shortest Path Problems * Dijkstra's Algorithm * Analysis of Dijkstra's Algorithm * Prim's Spanning Tree Algorithm * Summary	**6**
8.	Practice Lab (Team project 1) Quiz 1 实践/测验 1	* Lists and Recursion	**6**
9.	Practice Lab (Team project 2) Quiz 2 实践/测验 2	* Searching or Sorting	**6**
10.	Practice Lab (Team project 3) Quiz 3 实践/测验 3	* Trees	**6**
11.	Practice Lab (Team project 4) Quiz 4 实践/测验 4	* Graphs	**6**
Total Hours 总课时			**51**

Evaluation and Grading 考核方式与评估

Evaluation 考核方式	Percentage 百分比
Class Participation 课堂参与度	10%
Assignments & Labs 作业和实践	30%
Mid-term Exam 期中考试	20%
Final Exam 期末考试	40%
Total 总计	100％