目录

2025 年春季学期起 2

2024 年春季学期及之前 11

课程详述

COURSE SPECIFICATION

以下课程信息可能根据实际授课需要或在课程检讨之后产生变动。如对课程有任何疑问，请联系授课教师。

The course information as follows may be subject to change, either during the session because of unforeseen circumstances, or following review of the course at the end of the session. Queries about the course should be directed to the course instructor.

1.	**课程名称 Course Title**	**算法设计与分析（H） Algorithm Design and Analysis (H)**
2.	**授课院系 Originating Department**	计算机科学与工程系 Department of Computer Science and Engineering
3.	**课程编号 Course Code**	CS216
4.	**课程学分 Credit Value**	3
5.	**课程类别 Course Type**	专业基础课 Major Foundational Courses
6.	**授课学期 Semester**	春季 Spring
7.	**授课语言 Teaching Language**	中英双语 English & Chinese
8.	**授课教师、所属学系、联系方式（如属团队授课，请列明其他授课教师） Instructor(s), Affiliation& Contact （For team teaching, please list all instructors）**	陈杉，副教授，计算机科学与工程系，chens3@sustech.edu.cn Shan Chen, Associate Professor, Department of Computer Science and Engineering, chens3@sustech.edu.cn
9.	**实验员/助教、所属学系、联系方式 Tutor/TA(s), Contact**	
10.	**选课人数限额(可不填) Maximum Enrolment （Optional）**	

11. 授课方式 Delivery Method	讲授 Lectures	习题/辅导/讨论 Tutorials	实验/实习 Lab/Practical	其它(请具体注明) Other（Please specify）	总学时 Total
学时数 Credit Hours	32		32		64

12. 先修课程、其它学习要求 Pre-requisites or Other Academic Requirements	CS109 计算机程序设计基础 Introduction to Computer Programming CS203 数据结构与算法分析 Data Structures and Algorithm Analysis
13. 后续课程、其它学习规划 Courses for which this course is a pre-requisite	None
14. 其它要求修读本课程的学系 Cross-listing Dept.	Not applicable for other departments beside CS

教学大纲及教学日历 SYLLABUS

15. 教学目标 Course Objectives

This course introduces the fundamental concepts of algorithms through real-world applications and demonstrates how algorithms are used to solve practical problems in the field of computer science. It aims to help students understand the process of algorithm design and the importance of algorithms. By the end of the course, students will have mastered basic techniques for algorithm analysis, complexity theory, and a range of commonly used algorithms, such as greedy algorithms, divide-and-conquer algorithms, dynamic programming, network flow algorithms, and randomized algorithms. In addition to gaining theoretical knowledge, students are required to implement these algorithms using programming languages to enhance their practical skills.

本课程以真实世界的需求为切入点，讲解算法的基本概念，阐述如何在计算机领域中利用算法解决实际问题，帮助学生理解算法设计的流程及其重要性。通过本课程的学习，学生将掌握基本的算法分析方法、复杂度理论，以及一系列常用算法，包括贪心算法、分治算法、动态规划算法、网络流算法和随机算法等。此外，学生还需使用编程语言实现这些算法，从而巩固所学知识并提升实践能力。

16. 预达学习成果 Learning Outcomes

Upon completion of this course, students shall have the capabilities of doing:

1) Prove the correctness of algorithms.

2) Analyze the asymptotic order of growth of algorithms.

3) Be familiar with a series of fundamental algorithms such as greedy algorithms, divide-and-conquer algorithms, dynamic programming, network flow algorithms, and randomized algorithms.

4) Apply design and analysis methods to the above-mentioned major algorithms.

5) Implement these major algorithms by using a programming language.

6) Understand the algorithm design process.

7) Understand the role of algorithms in the broader field of computer sciences.

学习了本门课程后，学生可以具有如下能力：

1) 证明算法的正确性；

2) 分析算法计算量增长的渐进级别；

3) 熟悉一系列基础的算法，如贪心算法、分治算法、动态规划、网络流算法和随机算法等；

4) 对前述一系列算法能够进行设计和分析；

5) 使用某一种编程语言实现前述一系列算法；

6) 理解算法设计流程；

7) 理解算法在计算机科学领域中的作用。

17. 课程内容及教学日历 （如授课语言以英文为主，则课程内容介绍可以用英文；如团队教学或模块教学，教学日历须注明主讲人）
Course Contents (in Parts/Chapters/Sections/Weeks. Please notify name of instructor for course section(s), if this is a team teaching or module course.)

Week 1: Course Introduction and Basics of Algorithm Analysis

Computational Tractability

Asymptotic Order of Growth

A Survey of Common Running Times

Five Representative Problems on Independent Set

[Lab 1] Lab Introduction, Lab Techniques, Amortized Analysis, and solving simple pattern-finding lab problems

Week 2: A First Problem: Stable Matching

Definitions

Gale-Shapley Algorithm

Proof of Correctness

Efficient Implementation

Man Optimality and Woman Pessimality

[Lab 2] Designing stable-matching algorithms to solve lab problems

[Assignment 1] Analyzing an extended stable matching problem

Weeks 3 ~ 5 & Week 6 (1st lecture): Greedy Algorithms

Examples

Scheduling

Optimal Caching

Shortest Paths in a Graph

Minimum Spanning Trees

Single-Link Clustering

Min-Cost Arborescences

Huffman Codes

[Labs 3 ~ 5 & Lab 6A] Designing greedy algorithms to solve lab problems

[Assignment 2] Analyzing Chu-Liu's algorithm

Week 6 (2nd lecture) & Week 7: Divide-and-Conquer

Counting Inversions

Closest Pair of Points

Median and Selection

Integer and Matrix Multiplication

Convolution and FFT

[Lab 6B & Lab 7] Designing divide-and-conquer algorithms to solve lab problems

[Assignment 3] Surveying on FFT applications

Weeks 8 ~ 10: Dynamic Programming

Weighted Interval Scheduling

Segmented Least Squares

Knapsack Problem

RNA Secondary Structure

Sequence Alignment

Shortest Paths with Negative Weights

Distance-Vector Protocols

Negative Cycles

[Labs 8 ~ 10] Designing dynamic-programming algorithms to solve lab problems

[Assignment 4] Analyzing Tarjan's subtree disassembly trick

Weeks 11 ~ 13: Network Flow

Max Flow and Min Cut

Ford-Fulkerson Algorithm

Max-Flow Min-Cut Theorem

Capacity-Scaling Algorithm

Edmonds-Karp Algorithm

Dinitz's Algorithm

Bipartite Matching

Disjoint Paths

Extensions to Max Flow

Survey Design

Airline Scheduling

Image Segmentation

[Labs 11 ~ 12] Designing network-flow algorithms to solve lab problems

[Assignment 5] Analyzing Hungarian algorithm

Week 14 (1st lecture) & Week 16 (1st lecture): Randomized Algorithms

Content Resolution

Global Min Cut

Load Balancing

MAX 3-SAT

[Lab 13] Designing randomized algorithms to solve lab problems

Week 14 (2nd lecture) & Week 15: Computational Intractability

Polynomial-Time Reductions

Reductions by Simple Equivalence

Reductions from Special Case to General Case

Reductions via “Gadgets”

P vs. NP

NP-Complete

[Bonus Lab] Designing algorithms to solve bonus lab problems

[Assignment 6] Proving NP-completeness of some problems using reductions

Week 16 (2nd lecture): Review and Preparation for Final Exams

Course Review and Q&A

第 1 周：课程简介与算法分析基础

计算可处理性

增长的渐近阶

常见运行时间的概览

独立集的五个典型问题

【实验 1】实验简介、实验技巧、均摊分析以及解决简单模式匹配实验问题

第 2 周：第一个问题：稳定匹配

定义

Gale-Shapley 算法

正确性证明

高效实现

男性最优性与女性最劣性

【实验 2】设计稳定匹配算法解决实验问题

【作业 1】分析扩展稳定匹配问题

第 3~5 周 & 第 6 周第 1 讲：贪心算法

示例

调度问题

最优缓存算法

图中的最短路径

最小生成树

单链聚类

最小树形图

Huffman 编码

【实验 3 ~ 5 & 实验 6A】设计贪心算法解决实验问题

【作业 2】分析 Chu-Liu 算法

第 6 周第 2 讲 & 第 7 周：分治算法

计算逆序数

最近点对问题

中位数与选择问题

整数与矩阵乘法

卷积与快速傅里叶变换（FFT）

【实验 6B & 实验 7】设计分治算法解决实验问题

【作业 3】调研 FFT 的应用

第 8~10 周：动态规划

带权区间调度

分段最小二乘法

背包问题

RNA 二级结构预测

序列比对

带负权的最短路径

距离矢量协议

负权环检测

【实验 8 ~ 10】设计动态规划算法解决实验问题

【作业 4】分析 Tarjan 的子树拆解技巧

第 11~13 周：网络流

最大流与最小割

Ford-Fulkerson 算法

最大流最小割定理

容量缩放算法

Edmonds-Karp 算法

Dinitz 算法

二分图匹配

不相交路径

最大流的扩展应用

调查设计

航空公司排班

图像分割

【实验 11 ~ 12】设计网络流算法解决实验问题

【作业 5】分析匈牙利算法

第 14 周第 1 讲 & 第 16 周第 1 讲：随机算法

内容解析

全局最小割

负载均衡

MAX 3-SAT 问题

【实验 13】设计随机算法解决实验问题

第 14 周第 2 讲 & 第 15 周：计算不可处理性

多项式时间归约

通过简单等价的归约

从特例到一般情况的归约

通过“部件”的归约

P 与 NP 问题

NP 完全性

【附加奖励实验】设计算法解决附加奖励实验问题

【作业 6】通过归约证明某些问题的 NP 完全性

第 16 周第 2 讲：复习与期末考试准备

课程回顾与问答

18. 教材及其它参考资料 Textbook and Supplementary Readings

Jon Kleinberg and Eva Tardos, Algorithm Design, Pearson

课程评估 ASSESSMENT

19.

评估形式 Type of Assessment	评估时间 Time	占考试总成绩百分比 % of final score	违纪处罚 Penalty	备注 Notes
出勤 Attendance				
课堂表现 Class Performance				
小测验 Quiz				
课程项目 Projects				
平时作业 Assignments	3 hours per time	20%		
期中考试 Mid-Term Test				
期末考试 Final Exam	2 hours	40%		
期末报告 Final Presentation				
其它（可根据需要改写以上评估方式） Others (The above may be modified as necessary)	3 hours per week	40%		Lab Assignments 实验课作业

20. 记分方式 **GRADING SYSTEM**

☑ A. 十三级等级制 **Letter Grading**
☐ B. 二级记分制（通过/不通过） **Pass/Fail Grading**

课程审批 REVIEW AND APPROVAL

21. 本课程设置已经过以下责任人/委员会审议通过
This Course has been approved by the following person or committee of authority

课程详述

COURSE SPECIFICATION

以下课程信息可能根据实际授课需要或在课程检讨之后产生变动。如对课程有任何疑问，请联系授课教师。

The course information as follows may be subject to change, either during the session because of unforeseen circumstances, or following review of the course at the end of the session. Queries about the course should be directed to the course instructor.

1.	**课程名称 Course Title**	**算法设计与分析（H） Algorithm Design and Analysis (H)**
2.	**授课院系 Originating Department**	计算机科学与工程系 Department of Computer Science and Engineering
3.	**课程编号 Course Code**	CS216
4.	**课程学分 Credit Value**	3
5.	**课程类别 Course Type**	专业基础课 Major Foundational Courses
6.	**授课学期 Semester**	春季 Spring
7.	**授课语言 Teaching Language**	中英双语 English & Chinese
8.	**授课教师、所属学系、联系方式（如属团队授课，请列明其他授课教师） Instructor(s), Affiliation& Contact （For team teaching, please list all instructors）**	于仕琪，副教授，计算机科学与工程系，yusq@sustech.edu.cn Shiqi Yu, Associate Professor, Department of Computer Science and Engineering, yusq@sustech.edu.cn
9.	**实验员/助教、所属学系、联系方式 Tutor/TA(s), Contact**	赵耀，教学实验师，计算机科学与工程系，zhaoy6@sustech.edu.cn Yao Zhao, Assistant Teaching Technician, Department of Computer Science and Engineering, zhaoy6@sustech.edu.cn
10.	**选课人数限额(可不填) Maximum Enrolment （Optional）**	

11. 授课方式 Delivery Method	讲授 Lectures	习题/辅导/讨论 Tutorials	实验/实习 Lab/Practical	其它(请具体注明) Other（Please specify）	总学时 Total
学时数 Credit Hours	32		32		64

12. 先修课程、其它学习要求 Pre-requisites or Other Academic Requirements	CS102A 计算机程序设计基础 A Introduction to Computer Programming A CS203 数据结构与算法分析 Data Structures and Algorithm Analysis
13. 后续课程、其它学习规划 Courses for which this course is a pre-requisite	None
14. 其它要求修读本课程的学系 Cross-listing Dept.	Not applicable for other departments beside CS

教学大纲及教学日历 SYLLABUS

15. 教学目标 Course Objectives

The course aims to introduces basic concepts on algorithms by looking at the real-world problems that motivate them. It teaches various design and analysis techniques for problems that arise in computing applications, and encourage the students to understand the algorithm design process and the role of algorithms in the broader field of computer sciences. The students will be familiar with major algorithms, such as fundamental graph-based algorithms, greedy algorithms, divide-and-conquer algorithms, dynamic programming algorithms, and network flow algorithms. In addition, upon completion of this course, the students should be able to program these algorithms for solving corresponding problems.

本课程通过真实世界里的需求来介绍算法的基本概念，并介绍计算机领域内如何使用算法解决这些问题，以让学生理解算法设计流程和算法的重要性。学完本课程后，学生会掌握一系列常用的算法：图算法、贪婪算法、分治算法、动态规划算法、网络流算法、随机算法等。除了掌握基础知识之外，学生还需要使用编程语言实现这些算法。

16. 预达学习成果 Learning Outcomes

Upon completion of this course, students shall have the capabilities of doing:

1) Prove the correctness of algorithms.

2) Analyze the asymptotic order of growth of algorithms.

3) Be familiar with major algorithms like fundamental graph-based algorithms, greedy algorithms, divide-and-conquer algorithms, dynamic programming algorithms, network flow algorithms and some randomized algorithms.

4) Apply design and analysis methods to the above-mentioned major algorithms.

5) Implement these major algorithms by using a programming language.

6) Understand the algorithm design process.

7) Understand the role of algorithms in the broader field of computer sciences.

学习了本门课程后，学生可以具有如下能力：

1) 证明算法的正确性；

2) 分析算法计算量增长的渐进级别；

3) 熟悉一系列主要的算法：图算法、贪婪算法、分治算法、动态规划、网络流算法和随机算法；

4) 对前述一系列算法能够进行设计和分析；

5) 使用某一种编程语言实现前述一系列算法；

6) 理解算法设计流程；

7) 理解算法在计算机科学领域中的作用。

17. 课程内容及教学日历（如授课语言以英文为主，则课程内容介绍可以用英文；如团队教学或模块教学，教学日历须注明主讲人）
Course Contents (in Parts/Chapters/Sections/Weeks. Please notify name of instructor for course section(s), if this is a team teaching or module course.)

Week 1: Introduction to Algorithm Design and Analysis

Stable Matching Problem

[Lab 1] Reviewing the Stable Matching algorithm and construction skills of test cases

Week 2: Five Representative Problems

Interval Scheduling

Weighted Interval Scheduling

Bipartite Matching

Independent Set

Competitive Facility Location

[Lab 2] Programming stable matching algorithms

Week 3: Basics of Algorithm Analysis

Computational Tractability

Asymptotic Order of Growth

Common Running Time

[Lab 3] Writing a runtime survey program, and testing the algorithms with n, nlogn, n^2 and other time complexity. Observing that how the runtime changes with the increase of the input scale of the problem.

Week 4 &5: Graphs

Graph Traversal

Testing Bipartiteness

Connectivity in Directed Graphs

DAG and Topological Ordering

[Lab 4] Using BFS or DFS to solve some Graph Search Problems

[Lab 5] Using Topological Ordering to solve some common problems, such as courses scheduling or projects planning

Week 6 & 7: Greedy Algorithms

Interval Scheduling

Interval Partitioning

Scheduling to Minimize Lateness

Optimal Caching

Shortest Paths in a Graph

Minimum Spanning Tree

Clustering

Huffman Codes

[Lab 6] Solving some Interval Scheduling problems

[Lab 7] Using MST to solve some problems, using Huffman to solve some problems

Week 8 & 9: Divide and Conquer

Mergesort

Counting Inversions

Closest Pair of Points

Integer Multiplication

Convolution and FFT

[Lab 8] Implementing a simple application of Divide and Conquer

[Lab 9] Implementing a comprehensive application of Divide and Conquer

Week 10 & 11: Dynamic Programming

Weighted Interval Scheduling

Segmented Least Squares

Knapsack Problem

RNA Secondary Structure

Sequence Alignment

Shortest Paths

[Lab 10] Solving a simple dynamic programming problem; Solving the simple knapsack problem.

[Lab 11] Solving the comprehensive knapsack problem

Week 12 & 13: Network Flow

Minimum Cut Problem

Maximum Flow Problem

Ford-Fulkerson Algorithm

Good Augmenting Paths

[Lab 12] Explaining Network Flow algorithms and Application Examples

[Lab 13] Solving the comprehensive network problem

Week 14 & 15: Randomized Algorithms

Finding the Global Minimum Cut

Randomized Approximation Algorithm for MAX 3-SAT

Randomized Divide and Conquer

Hashing

Randomized Caching

[Lab 14] Contention Resolution

[Lab 15] Finding the closest pair of points by a randomized approach

Week 16: Review and Preparation for Final Exams

[Lab 16] Review, Q&A

第 1 周：介绍算法设计和分析

稳定匹配问题

[Lab 1] 学习稳定匹配问题，测试案例构建

第 2 周：算法中的五个代表性问题

区间调度

带权重的区间调度

二分图匹配

独立集问题

竞争设施选址问题

[Lab 2] 编程实现稳定匹配算法

第 3 周：算法分析基础

计算复杂性理论

渐进分析

常见运行时间

[Lab 3] 编写一个运行时间分析程序，测试具有 n、nlogn、n^2 和其他时间复杂度的算法，并观察随着输入量级增加，运行时间如何变化。

第 4-5 周：图算法

图遍历

测试双向图

有向图连通性

有向无环图及拓扑排序

[Lab 4] 使用深度优先或宽度优先解决图搜索问题。

[Lab 5] 使用拓扑排序解决一些常见问题，例如排课或者项目规划。

第 6-7 周：贪婪算法

区间调度

区间分割

最小延迟调度问题

贪心算法

图中的最短路径

最小生成树

聚类

哈夫曼编码

[Lab 6] 解决区间调度问题。

[Lab 7] 使用 MST 解决一些问题，使用哈夫曼编码解决问题。

第 8-9 周：分治算法

归并排序

统计逆序数

最近点对算法

整数乘法

卷积和快速傅立叶变换

[Lab 8]实现一个简单的分治算法的应用。

[Lab 9]实现一个完整的分治算法应用。

第 10-11 周：动态规划

带权重的区间调度

分段最小二乘法

背包问题

RNA 二级结构

序列对齐

最短路径

[Lab 10]解决一个简单的动态规划问题；解决一个简单的背包问题。

[Lab 11]解决一个完整的背包问题。

第 12-13 周：网络流算法

最小割问题

最大流问题

最大流量算法

好的增广路径

[Lab 12] 分析网络流算法和应用案例。

[Lab 13] 解决一个完整的网络流问题。

第 14-15 周：随机算法

寻找最小割

用于 MAX 3-SAT 的随机近似算法

随机分治算法

哈希

随机贪心算法

[Lab 14] 了解和分析竞争消除算法。

[Lab 15] 使用随机方法寻找最近点对。

第 16 周：复习本学期内容

[Lab 16] 复习、答疑。

18. 教材及其它参考资料 Textbook and Supplementary Readings

Jon Kleinberg and Eva Tardos, Algorithm Design, Pearson

课程评估 ASSESSMENT

19.

评估形式 Type of Assessment	评估时间 Time	占考试总成绩百分比 % of final score	违纪处罚 Penalty	备注 Notes
出勤 Attendance	5 min/per time	10%		
课堂表现 Class Performance				
小测验 Quiz				
课程项目 Projects				
平时作业 Assignments	2 hours/per time	20%		
期中考试 Mid-Term Test				
期末考试 Final Exam	2 hours	40%		
期末报告 Final Presentation				
其它（可根据需要改写以上评估方式） Others (The above may be modified as necessary)	2 hours per week	30%		Lab

20. 记分方式 **GRADING SYSTEM**

☑ A. 十三级等级制 **Letter Grading**
☐ B. 二级记分制（通过/不通过） **Pass/Fail Grading**

课程审批 REVIEW AND APPROVAL

21. 本课程设置已经过以下责任人/委员会审议通过
This Course has been approved by the following person or committee of authority