

## 南台科技大學 98 學年度第 2 期課程資訊

課程名稱	高等演算法
課程編碼	30M03801
系所代碼	03
開課班級	博研電子一甲 碩研電子一甲 海研電子一甲 碩研通訊一甲
開課教師	黎靖
學分	3.0
時數	3
上課節次地點	五 2 3 4 教室 P505
必選修	選修
課程概述	演算法是指利用電腦解決問題所需要的具體方法和步驟。也就是說給定初始狀態或輸入數據，經過電腦程序的有限次運算，能夠得出所要求或期望的終止狀態或輸出數據。本課程介紹電腦科學中重要的演算法及其分析與設計技術，熟知這些演算法，才能有效的使役電腦為我們服務。
課程目標	讓同學知道及學習撰寫好的程式，了解基本的演算法設計策略，知道這些策略有何優缺點及使用上的限制，並懂得經由複雜度分析判定演算法的好壞。
課程大綱	<p>第 1 章 演算法: 效率分析及複雜度</p> <p>1.1 概述</p> <p>1.2 幾個簡單演算法之範例</p> <p>1.3 演算法之效率</p> <p>1.4 逼近分析法</p> <p>1.5 複雜度分析</p> <p>第 2 章 分治法</p> <p>2.1 遞迴</p> <p>2.2 遞迴演算法之複雜度分析</p> <p>2.3 分治法</p> <p>2.4 分治法之限制</p> <p>第 3 章 動態規劃法</p> <p>3.1 概述</p> <p>3.2 費氏數列</p> <p>3.3 二項式係數</p> <p>3.4 最短路徑問題</p> <p>3.5 最佳的二元搜尋樹</p> <p>3.6 旅行推銷員問題</p> <p>3.7 動態規劃法之限制</p> <p>第 4 章 貪婪法</p> <p>4.1 概述</p> <p>4.2 最小代價開展樹</p>

	<p>4.3 單一起點最短路徑問題</p> <p>4.4 排程問題</p> <p>4.5 Huffman 碼</p> <p>第 5 章 回溯法</p> <p>5.1 概述</p> <p>5.2 n 后問題</p> <p>5.3 子集求和問題</p> <p>5.4 漢米爾頓迴路問題</p> <p>5.5 背包問題</p> <p>第 6 章 分支設限法</p> <p>6.1 概述</p> <p>6.2 背包問題</p> <p>6.3 分支設限之擇優搜尋法</p> <p>6.4 旅行推銷員問題</p> <p>第 7 章 時間複雜度: 排序問題</p> <p>7.1 插入排序法</p> <p>7.2 交換排序法</p> <p>7.3 選擇排序法</p> <p>7.4 簡單排序法之時間複雜度下限</p> <p>7.5 合併排序法</p> <p>7.6 快速排序法</p> <p>7.7 堆積排序法</p> <p>7.8 基數排序法</p>
英文大綱	<p>Chapter 1. Algorithms: Efficiency, Analysis, and Complexity</p> <p>1.1 Introduction</p> <p>1.2 Some simple algorithms</p> <p>1.3 The efficiency of algorithms</p> <p>1.4 Asymptotics</p> <p>1.5 Complexity analysis</p> <p>Chapter 2. Divide-and-Conquer</p> <p>2.1 Recursion</p> <p>2.2 Complexity analysis for recursive relation</p> <p>2.3 Divide-and-conquer</p> <p>Chapter 3. Dynamic Programming</p> <p>3.1 Introduction</p> <p>3.2 Fibonacci numbers</p> <p>3.3 Binomial Coefficient</p> <p>3.4 Shortest Paths Problem</p> <p>3.5 Optimal binary search trees</p>

	<p>3.6 The traveling salesperson problem</p> <p>Chapter 4. Greedy algorithm</p> <p>4.1 Introduction</p> <p>4.2 Minimum spanning trees</p> <p>4.3 Single-source shortest paths problem</p> <p>4.4 Scheduling</p> <p>4.5 Huffman code</p> <p>Chapter 5. Backtracking method</p> <p>5.1 Introduction</p> <p>5.2 The n-queens problem</p> <p>5.3 The Sum-of-Subsets Problem</p> <p>5.4 The Hamiltonian circuits problem</p> <p>5.5 Knapsack Problem</p> <p>Chapter 6. Branch-and-Bound</p> <p>6.1 Introduction</p> <p>6.2 Knapsack Problem</p> <p>6.3 Best-First Search with Branch-and Bound Pruning</p> <p>6.4 The traveling salesperson problem</p> <p>Chapter 7. Computational Complexity: The Sorting Problem</p> <p>7.1 Insertion Sort</p> <p>7.2 Exchange Sort</p> <p>7.3 Selection Sort</p> <p>7.4 Lower Bounds for Algorithms of Simple Sort Methods</p> <p>7.5 Merge Sort</p> <p>7.6 Quick Sort</p> <p>7.7 Heap Sort</p> <p>7.8 Radix Sort</p>
教學方式	實務操作,
評量方法	課堂討論,
指定用書	Foundations of Algorithms Using C++ Pseudocode
參考書籍	<ol style="list-style-type: none"> <li>1. T. H. Cormen, et al., Introduction to Algorithms, McGraw-Hill, 2002. (開發, 有中譯本)最受歡迎的教科書</li> <li>2. R. Sedgewick, Algorithm in C, Addison-Wesley, 1998. (新月,有兩冊)最好的教科書</li> <li>3. W. Ford and W. Topp, Data Structure with C++ Using STL, Prentice Hall, 2002.</li> <li>4. 張真誠、蔡文輝, 資料結構設計與 C++程式應用, 旗標.</li> <li>5. J. W. Crenshaw, MATH Toolkit for REAL-TIME Programming, CMP Books, ncku 005.131 C863.</li> </ol>

先修科目	
教學資源	R. Neapolitan and K. Naimipour, Foundations of Algorithms Using C++ Pseudocode, Jones and Bartlett Publishers, 2004. (新月代理)最深入淺出的教科書。
注意事項	
全程外語授課	0
授課語言 1	華語
授課語言 2	
輔導考照 1	
輔導考照 2	