

CH1、Recursive & Complexity

基本概念：遞迴與時間複雜度

目錄：

Algorithm

Recursive

定義、種類

與 Non-Recursive 比較

考題(程式與追蹤)

來源

7 種數學($n!$ 、 Σ 、費氏、二項式系數、Ackerman's、GCD、 x^y)

各章 DS(後續提及故先略過)

其他(河內塔、排列組合)

演算法定義

ADT

空間複雜度

Function Calling

Call by Value、Call by Address

時間複雜度

計算執行次數

Big O、Little O、Theta、Big Omega、Little Omega(比大小)

Recursive Time Function 計算

展開法

Master Method(case 3)、Extended Master Method(log n)

Recursive Tree(case 2)

[離散]特性方程式(費氏)

Guess+Proof

替換法

CH2、Array

陣列

目錄：

特質

計算

1~n 維

多項式表示法

Array、Linked List(CH4)

以係數存、只存非 0 項元素

特殊矩陣

稀疏矩陣

3-Tuple

上、下三角矩陣

對稱矩陣

寬帶矩陣

CH3 、Stack & Queue

堆疊與佇列

目錄：

Stack

定義、應用

ADT

製作

Permutation

push/pop(abc)、Binary Tree、括號、矩陣相乘、Y 字火車

中序轉前序、後序

括號法、Stack 演算法

轉換程式

後序、前序的運算

Stack 用於 Compile Parsing

$a^n b^n$ 、迴文、ab 出現次數是否相同

Queue

定義、應用

ADT

Queue 的種類(4 種)

一般 Queue、Priority Queue

Double-Ended Queue、Double-Ended Priority Queue

Stack, Queue 的互相轉換

Multiple Stack 製作

CH4、Linked List

鏈結串列

目錄：

- 定義、與 Array 的比較表、操作與配置
實作

 - Insert、Delete

 - 系統可用 Space 的管理

 - new, ret

- 利用 Linked List 製作 Stack

- 利用 Linked List 製作 Queue

 - Single Linked List, Circular Linked List, Double Linked List

 - Circular/Double 比較表

- Linked List 應用

 - 串列回收、長度計算、[補]Single→Double、串列合併、反轉串列

- 多項式表示法

 - Array(CH2：3 種方法)

 - Linked List：2 種方法

 - Single, Circular, Double Linked List

 - Generalize List

 - Generalize List 應用

 - Copy, Equal, Depth

 - Generalize List 之多項式表示法

CH5、Tree

樹與二元樹

目錄：

- 定義

- 表示法(4 種)

 - Linked List、二元樹、Child Sibling、括號法

- 二元樹

 - 比較表

 - 三定理

- 特殊二元樹

 - Skewed BT、Full BT、Complete BT、Strict BT

- BT 表示法

- BT 的應用

 - Traversal

 - 反向

 - Traversal 演算法

 - 應用演算法(7 個)

 - Count、Height、Leaf Node、Copy、Equal、SWAP BT、Expression

 - Binary Search Tree

 - Insert/Consturction、Search、Delete

 - Thread BT

 - Tree/Forest 與 BT 的轉換

 - Forest Traversal

 - [補充]n Node 可形成幾種 BT

 - Heap

 - Max-Heap、Min-Heap

 - Top-Down、Bottom-Up

CH6、Graph

圖論

目錄：

定義

術語(10 個)

Complete Graph、Subgraph、Path、Length、Simple Path、Cycle

Connected、Connected Component、Strongly Connected、Degree

表示方式

Adj Matrix、Adj List、Multi Adj List、Index+Array

Traversal(追蹤)

DFS、BFS

Spanning Tree

Minium Cost Spanning Tree

[補充]Union & Find(3 種方法)

Kruskal's、Prim's、Sollin's

Shortest Path Problem

	演算法	假設條件	時間複雜度	
One to all	Dijkstra's	邊的加權不可負值	$O(n^2)$	Greedy Algorithm
One to all	Bellman	允許有負值 但不可有負循環	$O(n^3)$	Dynamic Programming
All to all	Floyd-Warshall	允許有負值 但不可有負循環	$O(n^3)$	Dynamic Programming

A^+ , A^* 矩陣

AOV Network、AOE Network

Topological Order

Articulation Point、Biconnected Graph、Biconnected Component

CH7、Sort & Search

排序與搜尋

目錄：

Search

Linear Search

Binary Search

何時 Linear Search 較 Binary Search 來得更佳

Binary Search 的 Decision Tree

Interpolation(插補)

Sorting

依存放地點、依排序順序、依演算法

初階排序：

Insertion Sort

Binary Insertion Sort、Linear Insertion Sort 分別之改善

Selection Sort

Bubble Sort

Shell Sort

高階排序：

Quick Sort

改善 Quick Sort Worst Case 之方法(3 種)

Middle of Three

排序最快可達到多快之證明(Comparsion-Base)

Merge Sort

[Iterative 版]、[Recursive 版]

Selection Tree

Winner Tree、Loser Tree

Heap Sort

線性排序：

Radix Sort

LSD、MSD

Counting Sort

[重要]比較表

Selection Problem

CH8 、 Hashing

雜湊

目錄：

- Hashing Method

 - Why Hashing ?

- Hash Table 的結構

- Hash 中常見的問題

- Hashing 優點

- 良好的 Hash Design 應滿足

 - Perfect Hashing Function 、 Uniform Hashing Function

- 常見的設計方法

 - Middle Square 、 Mod(Division) 、 Folding Adding(2) 、 Digits Analysis

- Overflow 的處理方式

 - Linear Probing

 - Quadratic probing

 - Rehashing

 - Linked List/Chaining

 - [補充]：Double Hashing(演算法)

CH9、Advance Tree

高等樹

目錄：

Min-Max Heap(Insert, Delete)

Deap(Insert, Delete)

SMMH(Insert, Delete)

Extended BT

Weight External Path Length

Minimum Weight External Path Length

Huffman

如何評估 BST 優劣

AVL Tree

不平衡的調整方式

定理

m-way Search Tree

B-Tree

2-3 Tree

2-3-4 Tree(Insert, Delete)

B⁺ Tree

Red & Black Tree

OBST

Splay Tree

Leftist Tree

Min-Leftist Tree, Max-Leftist Tree(Insert, Delete)

Binomial Tree

Binomial Heap(Insert, Delete)

Fibonacci Heap

