

以色列高中電腦科學課程簡介

詹智傑 吳正己 何榮桂
國立臺灣師範大學 資訊教育系

摘要

隨著電腦的普及，許多國家已將電腦科學教育向下紮根，納入中等學校的教學科目中。以色列於 1990 年開始規劃新的高中電腦科學課程，其目標不只是教授學生如何操作電腦而已，而是一套嚴謹而與其它科學學科並列的獨立課程。其課程著重在電腦科學的核心觀念--演算法學，並強調理論與實務的結合，值得我們借鏡。

壹、前言

以色列的軟體產業發達，素有「海外矽谷」之稱，其軟體研發尤其精於影像處理及網際網路應用，亦吸引了全球著名的高科技企業，如微軟及英特爾，相繼在以色列設置研發中心，其中 Pentium II即是在以色列研製成功的(工商時報，1998.10.6)。由於以色列缺乏天然資源，因此將腦力激盪-軟體工業視為國家發展的關鍵產業，而無論在教育經費及企業研發經費佔國內生產毛額(GDP)的比例，皆為世界各國的前幾位(駐以色列代表處經濟組，1998)。以色列於 1994 年開始推動國家電腦化計劃(National Computerization Program)，期以五年的時間(1994-1998)，促使校園電腦化，提昇整個教育體系的資訊環境，並根據其實施成果，於 1999 年開始再以五年的時間進行第二個國家電腦化計劃(Pelgrum & Anderson, 1999)。

以色列教育部早於 1990 年即成立高中電腦課程發展委員會，將電腦科學規劃為高中教育中獨立的一門學科，地位等同於生物、物理、化學等科學學科，課程強調演算法學(algorithmics)為電腦科學的基礎，課程內容包含演算法的研討與分析、電腦的能力與限制、及電腦系統的設計與實作等(Gal-Ezer, Beeri, Harel, & Yehudai, 1995)。美國計算機學會(Association for Computing Machinery, ACM)亦於 1993 年提出高中電腦課程建議書(ACM Model High School Computer Science Curriculum)亦將電腦課程定位為基礎科學學科，課程旨在教授學生電腦科學的基本概念，期使學生能撰寫簡易程式、瞭解電腦的運作原理、及電腦的能力與限制等，建議修課期間為高一(十年級)全學年(Task Force of the Pre-College Committee of Education Board of the ACM, 1993)。而我國在 1996 年頒布新的高中電腦課程標準，電腦為選修科目，於高二或高三期間修習，課程旨在導引學生獲得電腦科學的基本概念與原理，培養學生應用電腦以解決問題的能力(吳正己、何榮桂，1998)。

台灣在硬體製造實力雄厚，而以色列則以軟體研發見長，其高科技產業之發達，在人才的培養方面應有值得參考之處。本文旨在介紹以色列高中電腦科學課程，他山之石，可以攻錯，期使國人能對電腦科學教育投以更多的關注。下文中若無特別註明，有關以色列之相關內容均參考 Gal-Ezer, Beeri, Harel 與 Yehudai(1995)及 Gal-Ezer 與 Harel(1999)二篇文獻。

貳、課程發展背景與理念

以色列的教育體系基本上與我國類似，採取中央集權式，教育部負責課程與教科書審訂、政策制訂、各級學校視導、教師任用、預算編定與教育發展等有關事項(丁志權，1995)。其學制主要為 6-3-3 制，即六年國小、三年國中、三年高中的 12 年免費教育；義務教育年限則為 10 年，至高中一年級。在三年的高中教育中，分必修學科和選修學科，每個學科一般分為 3 個或 5 個學習單元(study unit)，每個學習單元每週授課 3 小時，一學年共授課 90 小時，故完整修習一學科共需 270 小時(3 個學習單元)或 450 小時(5 個學習單元)。高中教育完成後學生可參加大學入學考試(matriculation exam)，依高中時修習的必修、選修學科進行學科能力的檢核，最低必須通過 6 個學科測驗，才能取得大學入學的資格，而電腦科學則是規劃為入學考試的選修學科之一。

以色列高中在 1970 年代即有電腦科學課程，當時是以 BASIC 程式語言教學為主，並非一完整而獨立的學科，且不受學校的重視。在 1980 年代，課程內容新增 Logo 及電子試算表語言(electronic spreadsheet language)單元，課程以電腦素養的教學為主。1990 年，以色列成立高中電腦課程發展委員會，將電腦科學規劃為高中教育中的一門科學學科，委員會成員由一位數學與電腦科學學者、三位電腦科學專家、二位高中電腦教師、及四位教育部教育專家，共 10 位所組成，其主要的任務不僅是課程的規劃，亦要負責未來教材設計、教師訓練、及實地試行課程等工作。

該電腦課程發展委員會認為「演算法學」實為電腦科學的核心所在，高中電腦科學課程不應只是程式語言的教學或應用軟體的操作，而應包含演算法的研討與分析、電腦的能力與限制、及電腦系統的設計與實作。而實施的方法則強調「拉鏈原則」(zipper principle)，即結合理論與實作的教學原則，一方面於課堂上講授理論概念，一方面將理論層次轉化為實際應用層次，兩者同時並行實施。以程式設計教學為例，即先於課堂上講授概念，再以紙上作業思考問題的解決方法，最後到電腦教室上機實作撰寫程式碼。

為實際發展課程，課程發展委員會委任數個專業的研究小組，每個小組由電腦專家、電腦科學教育學者、高中電腦教師共三至四人組成，針對不同的學習單元，一同合作，逐步完成詳細的課程大綱及教材內容。而初步的課程規劃在 1991 年即開始小規模實地試行，在實施過程中不斷地對課程大綱與教材內容進行增刪修改，以兼顧到理論與實際。整個課程已獲

其教育部認可，課程發展委員會仍繼續扮演監督者的角色，並隨時提供課程實施的建言。本課程將於 2001 年，於以色列全國實施（J. Gal-Ezer, email 連繫，2000 年 10 月 18 日）。

參、課程內容

前述的高中電腦課程委員會所規劃的電腦課程包含六個學習單元，單元名稱分別為：「基礎一」(Fundamental 1)、「基礎二」(Fundamental 2)、「軟體設計」(Software Design)、「理論」(Theory)、「第二種程式語言」(Second Paradigm)、「應用」(Application)，每一學習單元均授課 90 小時。課程共區分為三種方式供學生修習：

- (1)一個學習單元：為 90 小時的「基礎一」單元內容。
- (2)三個學習單元：必修單元為「基礎一」及「基礎二」，並自「第二種程式語言」及「應用」中擇一選修，授課共 270 小時。
- (3)五個學習單元：必修單元為「基礎一」、「基礎二」、「軟體設計」、及「理論」，並自「第二種程式語言」及「應用」中擇一選修，授課共 450 小時。

茲說明各學習單元之內容如下：

一、基礎一

內容為電腦科學的基礎概念(見表一)，高中一年級的學生均須修習。內容以基本演算法的設計與分析等概念為主，課程採用程序性語言—Pascal 為演算法設計的工具。課程中程式語言是學習演算法的工具，將演算法落實於程式設計中才是教學的主要目的。講課時數 60 小時，實習時數 30 小時。

表一 「基礎一」的學習主題

主題	講課時數	實習時數
Introduction	5	-
A basic computational model	9	6
Introduction to algorithm development	3	-
Conditional execution	5	4
Correctness of algorithm	3	1
Iterative execution	10	5
Efficiency of algorithms	3	-
Subtasks: Functions	3	5
One-dimensional arrays	7	5
Control structures revisited	12	4
總時數	60	30

二、基礎二

「基礎二」是延伸「基礎一」的學習內容，增加學生對演算法設計與分析的學習廣度及深度，內容涵括進階程式設計觀念，並觸及基礎資料結構(見表二)。講課時數 60 小時、實習

時數 30 小時。

表二 「基礎二」的學習主題

主題	講課時數	實習時數
Developing algorithms	8	-
Subtasks: Procedures	8	10
Types	2	2
Recursion	6	4
Characters and strings	2	4
Advanced algorithmic problems	18	6
Correctness and efficiency revisited	10	-
Introduction to data structures	6	4
總時數	60	30

三、軟體設計

內容主要是軟體系統設計的基本原則，包含資料結構的介紹及抽象資料型態(Abstract Data Type, ADT)的運用等，並觸及小型軟體系統的設計與發展方法（見表三）。講課時數 45 小時，實習時數 45 小時。（表三總時數之計算有誤，經與 J. Gal-Ezer 求證，她也未能提供正確數字。）

表三 「軟體設計」的學習主題

主題	講課時數	實習時數
Introduction	1	-
Library unit	2	4
Data types	5	7
Stack	3	3
Efficiency	7	4
List	12	9
Binary tree	11	6
Case study	6	4
總時數	45	45

四、第二種程式語言

課程發展委員會認為學習不同典範(paradigm)的程式語言，可以提供學生不同的思維方式與解決問題的方法，因此希望學生除了「基礎一」中的 Pascal 程序性語言外，再學習第二種的程式語言。目前已規劃出兩種程式語言供學生選擇：邏輯語言(logic language，見表四)與系統語言(system level language，見表五)。邏輯語言的設計是以 Prolog 為教學工具，介紹基本的

邏輯概念及知識表達方法，並包括遞迴、串鏈、樹等內容，講課時數 60 小時，實習時數 30 小時。系統語言以模擬 CPU 運作的軟體或 Turbo Assembler 為教學工具，介紹電腦系統的運作及組合語言，講課時數 45 小時，上機時數 45 小時。課程發展委員會希望往後能增加如物件導向(object oriented)或函數式(functional)等的程式語言供選修。

表四 「第二種程式語言」：「邏輯語言」的學習主題

主題	講課時數	實習時數
Propositional and predicate calculus	6	-
Logic programming at the prepositional level	2	2
Logic programming at the predicate level	2	2
Rules and inferences	3	2
Representing and formalizing knowledge	8	4
Input/output and numerical computations	2	2
Negation	4	2
Recursion	8	4
Compound data structures	3	2
Abstract data types for representing knowledge	4	-
Lists	8	4
Trees	3	1
Graphs	3	1
Project	4	4
總時數	60	30

表五 「第二種程式語言」：「系統語言」的學習主題

主題	講課時數	實習時數
Number systems	5	2
Computer organization	4	1
Organization and execution of programs	8	-
Basic concepts of assembly language	7	25
Assembling, linking and loading	8	-
The stack and subprograms	6	8
Interrupts	5	5
From high-level languages to assembly language	2	4
總時數	45	45

五、應用

內容主要是學習應用軟體，以因應未來專業上的需求。本學習單元強調電腦應用領域的理論與實作，一方面介紹軟體運作原理，一方面學習軟體實務操作。目前提供兩門選修：資

訊系統(information system，見表六)與計算機圖學(computer graphic，見表七)。資訊系統以套裝軟體 Microsoft Access 為教學工具，討論邏輯觀念的檔案結構及資料庫的設計與分析等，講課時數 45 小時，實習時數 45 小時。計算機圖學以圖學軟體為教學工具，圖學理論及實務操作並重。講課時數 45 小時，實習時數 45 小時。

表六 「應用」：「資訊系統」的學習主題

主題	講課時數	實習時數
Introduction	4	2
Relations	10	8
The relational data base	10	10
Data base management	3	-
Information systems	4	-
Data base design	6	5
Data flow diagrams	4	4
User interfaces	4	4
Project	-	12
總時數	45	45

表七 「應用」：「計算機圖學」的學習主題

主題	講課時數	實習時數
Introduction	3	17
Geometric models	8	4
The geometric data base	6	3
Representing curves and surfaces	6	4
From the data base to the screen	5	2
Transforms	7	2
Algorithms for image production	6	3
Information transfer between systems	4	2
Project	-	8
總時數	45	45

六、理論

本學習單元的目的是使學生接觸有關電腦科學理論相關的主題，內容包含計算模型(computational models)與數值分析(numerical analysis)。計算模型介紹自動機及正規語言等。數值分析介紹線性方程式、函數解題、誤差等觀念。此學習單元有兩種實施方式，一是計算模型 90 小時(見表八)，一是計算模型與數值分析各 45 小時，共 90 小時(見表九)。

表八 「理論」：「計算模型」的學習主題

主題	講課時數	實習時數
Finite systems	6	-
Deterministic finite automata	15	-
Regular languages	15	-
Variants of finite automata	15	-
Pushdown automata	12	-
The power of pushdown automata	12	-
Turning machines	15	-
總時數	90	-

表九 「理論」：「計算模型」與「數值分析」的學習主題

主題	講課時數	實習時數
1.計算模型		
Finite systems	4	-
Deterministic finite automata	10	-
Regular languages	10	-
Variants of finite automata	9	-
Turning machines	12	-
2.數值分析		
Basic concepts	6	2
An algorithmic approach to linear systems	11	6
Iterative solutions of non-linear equations	13	7
總時數	75	15

肆、師資培育

以色列的電腦課程發展委員會認為高中電腦教師應具有大學電腦相關科系學位，並受過正規的師資培育訓練。但由於合乎此資格的電腦教師普遍不足，故而在師資不足前僅能以調訓在職教師濟急。以色列的 Open University of Israel (OUI) 已提出一個電腦科學教師檢定學程，並為其教育部接受，此學程除用為在職教師訓練的主要內容外，並為大學電腦相關科系培訓未來電腦教師之參考依據。以下所述則為 OUI 後來所提出的修訂版(Gal-Ezer, 1995)。

該版本中，師資檢定課程內容由二個部份組成，即課程的修習及實習。實習部份與其它科目師資檢定的內容相同，此處僅描述電腦專業課程應修習部份。課程分為必修科目與選修科目，必修科目包括：

R1：A Pre-Programming Introduction to Algorithmics

R2：Introduction to Programming

R3 : Data Structures and Algorithmics

R4 : Automata and Formal Languages

R5 : Algorithmics

R6 : Selected Topics in CS Teaching

選修科目包含範圍較廣，以下僅列出與高中電腦課程內容較為相關的科目：

E1 : Computer Organization

E2 : Introduction to Information Systems

E3 : Introduction to Software Engineering (with ADA)

E4 : Prolog (or Lisp) and Selected Topics in Artificial Intelligence

E5 : Introduction to Numerical Analysis

E6 : Computer Graphics

無論是職前或參加在職訓練的電腦教師，都須修習上述所有必修科目。如果只要教授高中的一般電腦課程，僅須修習二個選修科目；如果要教授高中較深入或特殊領域的電腦課程，則須修習四個選修科目。此外，這些未來電腦教師都須修習二種不同類別的程式語言。

前述必、選修科目大部份與目前大學電腦科系所開設的科目近似，以下僅就較為不同的 R1、R5、及 R6 三個科目加以說明：

R1 : A Pre-Programming Introduction to Algorithmics

本科目的主要目的是教授演算法解題的概念，在不涉及程式設計的前題下，逐步的設計演算法，強調模組化、由上而下設計、逐步修正 (stepwise refinement) 等基本解題概念。

R5 : Algorithmics

本科目主要為整體電腦科學基本概念的介紹，這些內容是電腦科學教師所必須具備的。目前 OUI 所授的內容係以 Harel(1992)的教科書為依據，包括下列重要主題：演算問題 (algorithmic problem) 的輸出入定義及解題演算法、一般演算法的簡介、及其正確性與效率等的分析、電腦的限制 (如：intractability、undecidability)、及電腦與人類關係等。

R6 : Selected Topics in Computer Science Teaching

本科目的先修科目為 R1-R5 並修過 E3 或 E4 等科目。建議涵括的主題為：「電腦科學的歷史」，如由程式語言、作業系統、或演算法發展的觀點來闡述；「電腦科學的內涵及架構」，如：到底什麼是“電腦科學”？它是科學 (science)？工程 (engineering)？

或是科技 (technology) ? 「電腦科學課程」, 包括大學、高中的課程; 以及「電腦科學教學的相關議題」等。

伍、結語

以色列的軟體產業在世界佔有一席之地, 與其政府在政策上大力支持與推動有關; 其政府不僅在產業政策上的背書, 在學校資訊教育亦投注甚多心力。綜觀其高中電腦科學課程規劃, 必修時數最少為 90 小時(基礎一), 選修時數最多可達 450 小時(含基礎一), 紮實且深入。課程規劃精神係將電腦科學視為獨立的科學學門, 課程內容包含六個學習單元, 而每個學習單元內容均深廣兼具, 在教學上採用兼顧理論與實務的「拉鏈原則」, 可見其對高中資訊教育的用心與期待。相對地, 在我國高中教育階段, 電腦課程僅為一門選修科目, 並常被視為藝能學科, 地位不若其它科學學科, 授課時數僅 72 小時, 少於以色列甚多。隨著二十一世紀資訊時代的來臨, 為了厚實國家資訊基礎教育, 我國應可考慮將電腦科學列為高中階段重要的科學學門, 甚或是必修學科之一。

參考文獻

1. 丁志權(1995): 以色列小學教育制度。比較教育, 36 卷 1 期, 34-37 頁。
2. 工商時報(1998): 10 月 6 日。
3. 吳正己、何榮桂(1998): 高級中學新訂電腦課程的內涵與特色。科學教育月刊, 208 期, 26-32 頁。
4. 駐以色列代表處經濟組(1998): 以色列投資環境基本資料。工業投資簡訊, 192 期, 15-24 頁。
5. Gal-Ezer, J. (1995). Computer science teachers' certification program. *Computers & Education*, 25(3), 163-168.
6. Gal-Ezer, J., Beerli, C., Harel, D., & Yehudai, A. (1995). A high-school program in computer science. *IEEE Computer*, 28(10), 73-80.
7. Gal-Ezer, J. & Harel, D. (1999). Curriculum and course syllabi for a high-school program in computer science. *Computer Science Education*, 9(2), 114-147.
8. Mioduser, D., Nachmias, R., & Baruch, A. (1999). National ICT-related policies: Israel. In W. J. Pelgrum and R. E. Anderson (Eds.), *ICT and the Emerging Paradigm for Life Long Learning: A Worldwide Educational Assessment of Infrastructure, Goals and Practices* (pp. 51-54). Enschede: University of Twente, OCTO.

9.Task Force of the Pre-College Committee of the Education Board of the ACM (1993). ACM model high school computer science curriculum. *Communications of the ACM*, 36(5), 87-90.

化學反應

設計者：蕭次融

桌上的塑膠杯內有 10 支小滴管（編號 1~7），其中的 7 支小滴管內各含有一種下列化合物的水溶液，濃度約為 0.1M（莫耳濃度）。試自行設計實驗，寫出以五官即能分辨的所有反應的結果。正確的每一反應得 10 分，錯誤倒扣 4 分。另外有三支滴管 X、Y、Z 為未知溶液，各含有下列七種化合物中的一種，試一一檢驗之。

實驗的結果先紀錄於這一張紙的空白處或背面，最後才繕寫於答案紙上。答案的寫法，請參考答案紙上的例 1 與例 2。

七支小滴管的編號與內容物如下：

- | | | | |
|------|--------|-------|-------|
| 1.鹽酸 | 2.氯化鋇 | 3.硝酸銀 | 4.硝酸鉛 |
| 5.硫酸 | 6.氫氧化鈉 | 7.鉻酸鉀 | |

器材：

小滴管(含溶液)	10 支	蒸餾水	1 瓶
點滴板(黑與白)	各 1 張	面 紙	1 包
A4 白紙	1 張		

注意：

- 1.這是點滴試驗，要在點滴板或白紙上做。
 - 2.藥劑每次只用一、兩滴就足夠看出現象。
 - 3.藥劑用光了，不能補充，因此要先計劃好如何做才動手，以節省藥劑用量。
 - 4.在點滴板做好了實驗，用面紙擦乾，然後滴上蒸餾水數滴，再用面紙擦乾，就可再使用。
- （取材自：國立臺灣師範大學科學教育中心舉辦之臺北地區國中學生創意競賽題目）