

## 大學에 있어서의 컴퓨터科學 教育課程

李 禹 翰

### [1] 緒 言

최근 우리 나라의 大學卒業生이 전자계산 분야에 진출하는 경향이 뚜렷이 나타나고 있다. 特히 數學, 物理學, 經營學, 電子工學 등을 전공한 학생들의 진출이 顯著하다. 이런 경향에 비추어 이들 학생들로 하여금 적절한 준비과정을 거치게 하는 것이 요망된다.

近者에 數學과 計算 또는 計算機械 사이의 關係가 크게 변하였다. 현대적 전자계산기가 출현할 당초에는 복잡은 하지만 根本的으로는 無우틴한 계산을 하는 도구로서 電子計算機를 사용하였다. 그러나 內藏式 프로그램의 출현과 콤포넌트回路의 改良 등으로 因하여 電子計算機의 이론과 취급되는 문제의 길이와 그 다양성에 혁신적인 변화가 일어났다. 電子計算機는 금일 복잡한 각종 시스템의 시뮬레이션, 복잡한 논리적계산, 새로운 電子計算機나 電子計算機 조직의 설계에 사용되고 있다.

이와 같은 문제를 해결하고자 노력하는 동안 數值解析學, 프로그래밍 이론, 自動機械理論, 開閉理論, 콤비나토리알數學, 抽象代數學, 記號論理學 등을 포함하는 소위 컴퓨터科學이 形成되었다. 특히 電子計算機의 구조와 관련해서 생각하면 수학적인 推理의 특성이라고 할 수 있는 일반화 추상화 그리고 엄격한 논리성의 역할을 증시하게 된다. 따라서 컴퓨터과학의 연구에는 數學者일 필요는 없으나 數學者와 같이 사고할 줄 알아야 한다.

이와 같은 이유로 컴퓨터과학의 연구와 교육에 관심있는 대학은 수학자와 긴밀한 관계를 맺어야 할 것이다. 물론 대학에서 컴퓨터 서비스를 담당하는 자도 이에 참여하여야 한다.

### [2] 本論의 概要

본론은 다음의 세 종류의 과정으로 갈라서 논한다.

- ① 일반수학과목—컴퓨터 과학의 전문가가 되고자 하는 자를 위한 과목이다.
- ② 컴퓨터과학의 전문과목—이들은 표준적인 수학과목의 범위 외에서 강의한다.
- ③ 컴퓨터과학 입문과목—일학기 3 학점의 강의로서 학생들로 하여금 高速 算

자계산기의 능력과 그 한계성을 인식하게 한다.

특정된 言語를 습득케 해서 프로그래밍을 작성하여 실지로 전자계산기로 처리한다. 이 과목은 컴퓨터과학의 기본과목으로서 모든 전문과목의 先修과목이다.

다음 4에서는 수학을 전공하는 자로서 컴퓨터 과학분야에 진출하고자 하는 자를 위한 과정을 취급한다. 이것은 數學科의 표준과정을 적절하게 수정한 것으로서, 기하학이나 위상수학은 제외되어 있고, 확률과 수치해석학이 중시된다. 컴퓨터 전문과목 중에서 적어도 2과목을 선택하여야 한다.

다른분야에서 컴퓨터과학에 진출하고자 하는 자를 위한 과정은 5에서 언급한다. 이러한 자를 위해서는 적어도 21학점의 수학과목을 이수케하고 각자의 專攻을 고려하여 컴퓨터과학 과목 중에서 적당수의 학점을 취득케 한다.

### [3] 컴퓨터科學의 入門科目

일상생활과 교육에 있어서 전자계산기와 같은 현대적 자동기계는 점점 그 중요성이 증가할 것이므로, 大學의 學部 학생은 일찍부터 이에 대한 이해와 그 사용 방법을 알아두는 것이 좋다. 따라서 3학점의 컴퓨터과학 입문 과목이 提供되어야 한다. 이 과목은 다음과 같은 의의가 있다.

- (1) 학생으로 하여금 전자계산기의 능력과 그 한계성을 인식하게 한다.
- (2) 기계, 사용언어, 문제의 산법적 형성 사이의 관련의 이해를 발전시킨다.
- (3) 현대적 전자계산기의 사용 훈련을 한다.
- (4) 後에 이수하는 과목에 있어서의 문제를 전자계산기를 이용하여 해결할 수 있다.

이 과목은 數學 教授가 강의할 수도 있으나 표준적인 수학 교과 과정의 하나로 국한시키는 것은 아니다. 컴퓨터 과학의 기본과목으로서 현대적 전자계산기에 흥미있는 모든 학생에 대한 입문 과목이다.

이 과목은 다음의 세단계로 나누어서 취급할 수 있다.

Ⓐ 전자계산기의 概要...여기서는 전자계산기의 논리적 구조와 산법을 다룬다. 二項 연산의 四則과 전자계산기의 부분의 기능을 설명한다.

Ⓑ 프로그래밍言語의 통용 규칙은 콤파일라를 통해서 얻은 機械語와 관련시켜서 그 구조를 알아 본다.

Ⓒ 문제의 해결...여기서는 소위 반복법에 의해서 많은 문제가 해결된다는 점을 강조한다. 流通圖의 개념을 소개하여 이용케 한다. 수학적인 문제의 경우에

는 근사법의 성질도 아울러 논하여야 한다. 기계로 처리할 연습 문제는 비 수치적인 것과 수치적인 것을 모두 課하고, 기계 사용에서의 능률과 문제 해결 방법의 다양성에 역점을 둔다.

[4] 數學專攻系를 위한 과정

다음의 과정은 수학 전공자로서 컴퓨터과학 분야로 진출하고자 하는 자를 위한 것이다. 과목 1, 2, 3, 4, 5는 수학전공의 핵심이며, 총요구 학점수는 각 대학에 따라 다소 차이가 있을 수 있다.

㉔ 컴퓨터과학의 입문 과목은 필수과목이다. (3 학점)

㉕ 필수수학과목

1. 微積分學(12學點)
2. 線型代數學(3 ")
3. 確率과 統計(3 ")
4. 代數의 構造(3 ")
5. 解 析 學 (6 ")
6. 數值解析學 (3 ")

㉖ 선택 과목(각 3 학점)...최소 6 학점을 요한다. 컴퓨터과학 입문과목은 다음 과목들이 先修 과목인데, 과목 8(논리학)만은 예외일 수 있다.

7. 數值解析學
8. 論 理 學
9. 機械의 組織
10. 情報의 處理
11. 自動機械理論
12. 고급 프로그래밍
13. 콤비나토릭스
14. 시스템 시뮬레이션

[5] 他專攻者를 위한 과정

수학을 제외한 자 분야에 요구된다. 이런 자를 위해서는 다음의 과목을 들 수 있다.

1. 微 積 分 學(12 학점)
2. 線型代數學 ( 3 " )
3. 確率과 統計( " )

## 4. 代數의 構造( " )

## [6] 敎課內容

## 1. 微積分學(12 학점)

## a. 微積分의 基礎概念

## b. 1 變數函數의 微積分의 이론과 응용

## c. 無限級數의 級數展開

## d. 多變數函數의 微積分

e. 常係數線型微方, 齊次, 非齊次에서의 線型代數의 응용, 1階線型, 非線型의 Picard 방법과 수치방법

## f. 複素數函數의 소개

## g. 2次元, 3次元에서의 벡터函數의 微積分

## h. 수치계산의 기교와 이론은 적시에 취급

## 2. 線型代數學(3 학점)

유한차원 벡터공간에서의 대수와 기하, 선형변환, 행렬대수, 고유치, 고유벡터

## a. 實數體 위의 유한차원벡터공간(5 시간)

## b. 선형변환과 행렬(7 시간)

c. 1차연립방정식(行列式, 小行列式) rank, adjoint, elementary transformation, 역행렬의 계산(11 시간)

## d. 行列의 他과학에의 응용(3 시간)

e. 2次形式(내적, 길이, orthogonal project) Gram-Schmidt 對角行列化, 陽值形式

f. 고유치, 고유벡터(특성다항식과 2차행렬식의 대각행렬화와의 관계, 특히 대칭행렬에서의 고유치, 고유벡터의 계산법)

## 3. 確率과 統計(3 학점)

## a. 確率(discrete : 17 시간, continuous : 9 시간)

sample space, event, random variable 과 그 함수, probability variance, moment, Chebychev 부등식, joint-distribution, joint-densities 의 transformation, conditional probability, Bayes theory, independence, Bernoulli trial, combinatorics, binary distribution, Poisson distribution, normal law, law of large numbers, central limit theory, elementary Markov chains.

b. Statistical Inferences (13 시간)

통계문제의 형식과 통계적 procedure 의 선택에 관한 고려, estimation, sampling, hypothesis testing, power of test, regression, non parametric 방법

4. 代數構造 (3 학점)

群, 環, 體의 표준적 취급

5. 解析學 (6 학점)

실변수함수론의 입문으로서 대상학생에 따라서 응용면의 비중을 적절히 조절한다.

6. 數值解析學 I (3 학점)

微分積分學의 전반이 先修, 선형대수는 동시 이수 가능

a. 방정식의 해법(non-linear 의 반복법, 연립방정식의 반복법, 복소계수 방정식) (6 시간)

b. 다항식 근사법(보간법, 구적법, Weierstrass, Bernstein, Lagrange, Chebyshev, Gauss 異常적분법) (18 시간)

c. 常微方의 초기치 문제(reduction to 1st order system, Runge-Kutta, Adams, predictor corrector methods, round-off, stability) (19 시간)

d. 역행열, 고유치(Gauss 소거법), 반복법 역행열 계산, Rayleigh quotient, power methods(6 시간)

7. 數值解析學 II (3 학점)

해석학, 선형대수학, 수치해석학 I 이 先修

a. 역행열과 고유치(복습, 3 각형행열, Given-Heusenholder 방법, 반복법) (10 시간)

b. 常微方, 경계치문제, 고유치문제(finite difference 방법), energy 方法, min-max(11 시간)

c. 2階偏微方(分類, 解析的解, 타원, 포물형, maximum-principles,  $L_2$ -or energy estimates, pointwise estimate, 쌍곡형, domain of dependence, Fourier 해석, 상계수 방정식에 대한 stability, 타원형에 대한 고유치, 편미방에 나오는 difference 方程式에 대한 반복법) (18 시간)

8. 論理學(3 학점)

deductive logic 과 computability theory 의 입문

a. propositional calculus with equality(10 시간)

- b. Quantification Theory(minimization theory) (10 시간)
  - c. Application (9 시간)
  - d. Turing machine, Post production system (10 시간)
9. 情報處理(3 학점)
- 대량의 수치, 비수치정보처리의 기술과 개요. a, b, c는 사무데이터에 관한 것이며 전체의 2/3. d, e는 자연언어로 된 정보취급
- a. 파일처리(파일書式, 주변장치의 특성, 파일처리언어)
  - b. Search-Sort(파일의 Search-Sort 방법, random access 파일, multi-level memory complexes의 사용)
  - c. Application(data reduction, analysis of written text, 파일 maintenance, document preparation, real time application)
  - d. Automatic Translation
  - e. Information Retrieval
10. Machine Organization (3 학점)
- 컴퓨터 system의 logical design에 관련된 개념의 입문
- a. 컴퓨터 기능의 복습
  - b. 數(변수, 4 則)
  - c. Boole 代數(집합론적—논리적취급, 眞理表, Boole 함수, 표준형)
  - d. Combinatorial Circuit(Switches, relay, Solid state devices)
  - e. Sequential machine(state diagram, equivalence, digital computer element, adders code converters, shift registers, accumulators)
  - f. logical design of a simple digital computer
11. Theory of Automata (3 학점)
- 代數의 構造(分割, 동치, 귀납법, 동형, 자기동형, 표준형 등)가 先修
- a. 유한 Automata의 정의
  - b. 유한 Automata의 유용성의 한계, Kleene의 정리
  - c. Sequential machine의 reduced form
  - d. 유한 Automata의 algebraic description(半群, 半群의 分割, monomorphism)
  - e. 他 class의 machine (Turing machine, non-deterministic machine, probabilistic machine)
  - f. Theory of Computability

g. Self-producing & Self-repairing machine, Von Neumann & Myhil의 이론

12. 高級 프로그래밍(3 학점)

a는 2 주, b, c는 13 주

a. 복습, assembly system, storage allocation 방법, pseudo-orders, macros, modify, load technique, monitor, executive system

b. 언어의 구조(Algol, recursion 과 procedure 를 취급하는 方法, list processing 언어, compiler writing 언어)

c. Theory of Compiler(syntax 지향 compiler, 문제지향 compiler, compiler syntax 언어를 취급하는 compiler)

13. Combinatorics (3 학점)

群論, 대수구조에 이용될 내용으로서 G. Polya의 기본정리를 이해케 한다.

a. 순열조합(집합론, 술어와 生成함수, Ryser, Riordan)

b. principles of inclusion and exclusion(Riordan, Ryser)

c. 生成함수, 합성 함수의 derivatives

d. permutation의 cycle

e. enumerative combinatorial analysis의 기본정리(G. Polya)

14. System simulation(3 학점)

operating system의 model로서 컴퓨터를 사용하여 실제와 비교 실험한다.

a. Simulation 用의 언어(DYNAMO, SIMSCRIPT, GPSS- II)

b. Simulation의 기술적 문제(synchronization of events, file maintenance, random number generation, random deviate sampling)

c. Simulation에 특이한 통계적 문제

(sample size estimate, variance reducing technique, problem of drawing inference from a continuous stochastic process)

d. Applications

(queuing models; storage, traffic and feed-back system; design of facilities and operating disciplines)