



06

東京大学における 全学プログラミング教育

森畑明昌 (東京大学大学院総合文化研究科)

東京大学における 1・2 年生教育

東京大学での教育で、最も特徴的なことは何でしょうか？ 1つの答としては、「1・2年生は、理系も文系も全員教養学部にも所属し、全学的なリベラル・アーツ教育を受ける。各学部での専門教育は2年生の後半から始まる」ことが挙げられるでしょう。1・2年生は語学や実験などの必修科目以外は、好きな科目を自由に履修できるのです。

1・2年生向けプログラミング教育も、この全学教育の枠組みのもとで行われています。東京大学は総合大学ですから、受講者には情報系学科を志望する学生もいれば、農学部や文学部などを志望する学生もいます。教員の側から見ると、スキルもモチベーションもまったく異なる学生数百人、または数千人に、一様にプログラミングを教えなければならない、ということになります。本稿では、この状況に対処するための工夫を中心に、本学の全学プログラミング教育についてお伝えします。なお、本稿の内容は筆者個人の見解であり、所属組織を代表するものではありません。

全学プログラミング教育の概要

プログラミングに関する1・2年生向け講義のうち主要なものを表-1に示します。特に「情報」と「ア

ルゴリズム入門」の2つが中心的な役割を担っています。両講義の内容を簡単に紹介しましょう。

情報

「情報」は1年生全員が5セメスター（夏学期）に履修する必修科目で、2006年度から実施されています。履修者数は3,000名以上、約30クラスを開講しています。

講義の目的は「情報リテラシー」を学ぶことです。これは、パソコンや応用ソフトウェアの使い方ではありません。コンピュータや情報システムがどう動作しているのか、情報技術が社会に与える経済的・倫理的影響、情報を表現する上での表現技法や法律として留意すべき点、など、情報技術が発展し変化したとしても変わらず必要となる普遍的な知識を学ぶことを目標としています。具体的な内容は、情報の表現、情報通信、アルゴリズム、情報システムの仕組み、インタフェース、情報技術と社会とのかわり、などです。

プログラミングについても「コンピュータによる問題解決」や「コンピュータの仕組み」との関連で学びます。プログラミングを学ぶ意義は大きく2つあります。まず、問題を分析し解法を表現するための、問題解決のための言葉としての役割です。もう1つは、コンピュータの動きを体感するためのツールとしてです。これらは、コンピュータや情報シス

情報	全学生必修。情報・情報処理に関する基礎を概観する。
アルゴリズム入門	理系準必修。プログラミングを通じ情報科学の基礎を学ぶ。
計算機プログラミング	経験者向け。基本的なアルゴリズムとデータ構造について学ぶ。
実践的プログラミング	経験者向け。ACM ICPC プログラミングコンテストを題材に発展的内容を扱う。

表-1 東京大学の主な1・2年生向けプログラミング関連講義 (2015年度)



テムの動作を理解し活用するには欠くことのできないものでしょう。なお、講義のうちどの程度をプログラミング関連の話題に割くかは担当者次第ですが、2～3週程度の方が多くようです。

アルゴリズム入門

「アルゴリズム入門」はAセメスター（冬学期）の選択科目で、2006年度から実施されています^{☆1}。理系の1年生には、いわゆる「準必修」となっています。履修者数は600～900名程度、開講クラス数は12ほどです。これの履修者数は本学の理系1年生の約1/3～1/2にあたります。もちろん、多くはありませんが文系の履修者もいます。

講義は大きく前後半に分かれています。前半ではプログラミング入門として、繰り返し、条件分岐、再帰などを学びます。後半ではプログラムを書きながら、アルゴリズムと計算量、数値計算法と数値誤差など、情報科学・問題解決で重要な概念を学びます。具体的には、フィボナッチ数の高速な計算、整列アルゴリズム、数値積分、疑似乱数、文字列の類似度計算などを扱います。プログラミング環境はMac OS X上のRubyです。

この講義の力点は後半、つまり情報科学の基礎の理解にあり、その手段としてプログラミング演習を行っています。もちろん、全学教育なので、それほど高度な内容に踏み込むつもりはありません。計算量と数値誤差を知ることが主な目標です。モンテカルロ法や動的計画法などのやや高度な内容もあるのですが、これらはあくまでも発展的な話題としての扱いです。基本的には、高度な内容は既習者向けの講義や情報系学科での教育に任せています。

実施上の困難と工夫

「情報」や「アルゴリズム入門」にはいくつかの難しさがあります。

まず、総合大学の全学教育であるため、コンピュータやプログラミングに興味のない学生、苦手

^{☆1} 2014年度までは「情報科学」という題目でした。

意識を持っている学生が多くいます。「プログラミングは私の将来に必要な、なぜ学ばなければならないのか」と考える学生も少なくありません。そのような学生にも、プログラミングの重要性と面白さを伝えなくてはなりません。

次に、受講者のレベルは本当にまちまちです。パソコンをほとんど触らない学生もかなりいる一方、すでにプログラミング経験のある学生もいます。そんな中、初めての学生が落ちこぼれないよう、経験者は退屈しないよう講義を進めなければなりません。特に問題になるのは成績です^{☆2}。プログラミングスキルで成績をつけるのでは本来講義を受けてほしい初学者が敬遠しかねません。

また、プログラミング演習のためには履修者数に比例する教員が必要になる、という問題もあります。特に「情報」では、履修者数が膨大であるため、プログラミングが専門ではない教員も講義を担当せざるをえません。そのような教員の多様性が学生にとって不利益や不公平にならないようにしなければなりません。

以上の問題を踏まえ、本学では次の工夫を行っています。

プログラミングはツールである

最も大きな工夫は、講義の設計思想です。「情報」「アルゴリズム入門」ともに、プログラミングができるようになることは主たる目的ではありません。あくまで学習のための手段と位置付けています^{☆3}。

試験や課題でも、プログラミングスキルよりは、情報科学の基礎を理解しているかどうかにかんして重きを置いています。これは、英語の講義の成績を、単語や文法のテストではなく、英語文献の輪講と最終レポートでつけるようなものかと思えます。英語が苦手な学生にとっては、この方が努力が成績につながると思えるのではないのでしょうか。

^{☆2} 本学では進学先の決定に成績を用いるため、成績の公平性は非常に重要です。

^{☆3} この点は、「情報」の前身である「情報処理」、「アルゴリズム入門」の前身である「計算機プログラミング」と比べての大きな違いです。「情報処理」はコンピュータの操作法に、「計算機プログラミング」はプログラミング入門に、それぞれ比較的重きを置いていました。

この立場を明確にしていることで、プログラミングに恐怖感を持つ学生にも多少とつきやすくなり、結果的にプログラミングのできる学生を増やしていると考えています。一方で、プログラミング経験のある学生にとっても、単なるプログラミング以上の知識を求められるため、達成感はあると思います。

プログラミングは「語学」である

プログラミングなんて自分には無理ではないか、そう思う学生は少なくありません。これに対しては、「プログラミングは外国語などと同じ『語学』。最初はちんぷんかんぷんでも、やれば誰でもできるようになる」とアピールしています。特に、プログラミング演習の側面が強い「アルゴリズム入門」では、「それほど難しくない演習課題をこなしていけば、自然に単位が取れる」という設計にし、学生にもそう伝えています。

実際、プログラミングの障壁はできる限り下げています。これは、プログラミングは手段、という趣旨にも合致します。Rubyを採用したのは、対話的環境のおかげで動きを理解しやすく、日本語の資料が充実しており自習しやすいからです。しかし、Ruby特有の機能は使いません。アルゴリズム入門の必修範囲では、変数はローカル変数だけ、制御構造はif, for, while, 関数のみ、あとは配列ぐらいです。オブジェクト指向はおろかreturnも扱いません。二次元配列の生成など、面倒な処理にはサポートプログラムを配布しています。扱うプログラムも極力シンプルなものにとどめています。多くは1関数あたり10行以内、複雑なものでも20～30行程度、といったところです。

「プログラミングはやれば誰にでもできる」という認識が学生に広がったのか、「アルゴリズム入門」の履修者数は増加傾向にあります。



図-1 川合慧編「情報」
東京大学出版会（2006年）

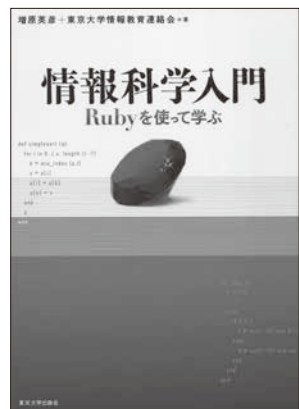


図-2 増原英彦+東京大学情報教育連絡会著「情報科学入門 Rubyを使って学ぶ」
東京大学出版会（2010年）

標準化と多様性のバランスをとる

多数のクラスを開講しなければならない、という問題には以下のような方針でのぞんでいます。

1. 講義資料は共通のものを用意しています。教科書は専用のもの（「情報」は図-1、「アルゴリズム入門」は図-2）を用意していますし、加えて講義スライド、演習課題、補助プログラムなども整備しています。
2. 必修範囲は共通講義資料（特に教科書）上で指定し、全クラス共通の試験を行います。
3. 以上を踏まえてどのような講義を行うかは各担当者に一任します。

まず、さまざまな先生が講義を担当するという状況を踏まえ、ベースとなる講義資料を充実させています。これは過去に各担当者が使っていたものを集約したもので、現在では多くの担当者がこれを自分のスタイルに合わせて修正しつつ使っています。次に、クラス間の公平性のため、試験範囲は学生に明示して共通の試験を行っています。一方で、公平性を気にするあまり、講義が画一的なものになり、各担当者の強みや独自の工夫などが生きなくなってしまっは本末転倒です。そのため、講義の進行や内容の選択含め、試験以外については各担当者に任せています。成績上の公平性を確保しつつ内容を一任するために、必修範囲や共通試験問題を準備している、ともいえます。

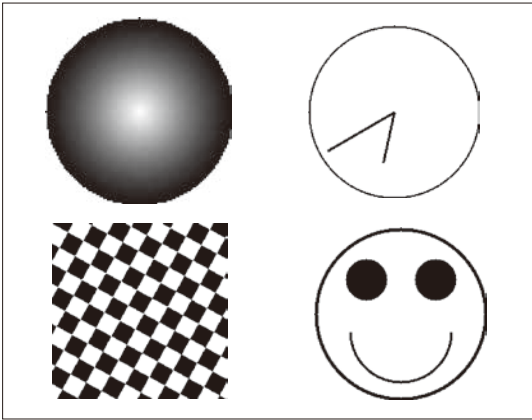


図-3 isrb による画像の例

動く・見える楽しさを大事にする

せっかくのプログラミングなので、自分が書いたものが動くのを楽しんでもらいたいものです。このため、「アルゴリズム入門」では isrb^{☆4} という Ruby の拡張処理系を用いています。isrb では簡単にビットマップ状の画像を表示することができます。図-3 に isrb で作成した画像の例を示しました。左上は教科書(図-2)にある例で、簡単な2重ループで記述できます。残り3つは学生の提出物をもとに作ったものですが、いずれも20~30行ほどの小さなプログラムです。このように、多少のプログラムで案外見栄えのする画像を作ることができるため、学生には良い動機付けになっていると感じています。また、計算量の講義では、プログラムの実行時間を自動でプロットする補助プログラムを用意し、アルゴリズム間の速度差が目に見えるようにしています。

応用へのつながりを忘れない

プログラミングはともすれば机上の話に終わがちです。机上のことが全世界につながる、という情報技術の魅力を知ってもらうため、応用との関連は

☆4 <http://prg.is.titech.ac.jp/i2cs.rb/isrb2/>

折に触れて語るようにしています。たとえば、連立一次方程式の解法を学ぶときにはそれがCTスキャナで使われていることを、動的計画法を学ぶ際にはそれがDNAの類似度計算に使えることにふれています。このような説明は、比較的プログラミングに縁遠い、生物系などを志望する学生の興味を引くようです。

.....
これからのプログラミング教育に向けて
.....

以上述べてきたようなかたちで、本学では1・2年生向けの全学プログラミング教育を行っています。現状の体制になって10年が経ちました。工夫として上手くいったものもありますが、まだまだ問題があると思える部分もあります。

- 情報技術は変遷が早い。共通資料中の話題や演習用のシステムなどで、すでに古めかしくなってしまったものが少なくない。
- 各担当者が個別に行っている工夫を集約・再利用することができていない。
- 最近の学生は一昔前の学生に比べプログラミングに対する興味が薄い。より「今の学生に合った」動機付けの方法が必要。

これまでの経験も踏まえ、さらなる改善の方策を練っているところです。

本学の状況が皆様のご参考になれば幸いです。

(2015年12月4日受付)

森畑明昌 (正会員) ■ morihata@graco.c.u-tokyo.ac.jp

2009年東京大学大学院情報理工学系研究科博士後期課程修了。同年日本学術振興会特別研究員、2010年東北大学電気通信研究所助教を経て、2014年より東京大学大学院総合文化研究科講師となり現在に至る。博士(情報理工学)。プログラミング言語の基礎理論に興味を持つ。

