

アルゴリズムとプログラミング

中西 通雄¹⁾ 佐々木 整²⁾ 高橋 等³⁾

1. はじめに

高等学校学習指導要領(平成 30 年告示)では、必履修教科「情報 I」におけるプログラミングに関して身に着けるべき項目が次のように記載されている(筆者による抜粋)。

- (3)ア(イ) アルゴリズムを表現する手段、プログラミングによってコンピュータや情報通信ネットワークを活用する方法について理解し技能を身に着ける
- (3)イ(イ) 目的に応じたアルゴリズムを考え適切な方法で表現し、プログラミングによりコンピュータや情報通信ネットワークを活用するとともに、その過程を評価し改善する
- (3)イ(ウ) 目的に応じたモデル化やシミュレーションを適切に行うとともに、その結果を踏まえて問題の適切な解決方法を考える

周知のとおり、この学習指導要領は令和 4 年度から実施される。また、令和 7 年度大学入学共通テストの出題教科に「情報」を加え、「情報 I」を出題科目とすることが、本年 7 月 30 日に文科省から正式に発表された。上記のとおり、プログラミングは「情報 I」の学習項目になっていることから、令和 7 年度からの大学新入生に対する情報教育においては、プログラミングを学習していることを前提にできることが期待される。なお、プログラミング言語の学習が目的でないことに留意しておく必要がある。

さて、私立大学情報教育協会において、学士力としての「情報活用能力」の一部としてのプログラミング教育の内容を考えるにあたり、学習者の知識・技能としてどの程度の前提を置くかを考えねばならない。本稿を執筆している時点では、高校の「情報 I」の教科書が出版されておらず、高校でのプログラミング教育の実態もこれからどのように進むかも不透明である。また、文科省が Web サイトに公開している教員研修用資料があるが、その内容は高校生が学ぶ内容よりも高度であるため、参考程度にとどめるのが適切である。そこで、次のとおり前提を置くことにした。

- 1) どの言語でもよいので、例えば次のようなプログラムを作成できる
 - ◇ 西暦年を入力して、うるう年かどうかを判定する(条件分岐)
 - ◇ 1から100までの整数のうち3または5の倍数の和を求める(繰り返し+条件分岐)
- 2) 配列の処理(例えば順次探索)や、関数の定義と呼び出しについては、大学入学前の前提スキルにするのは厳しい
- 3) 学んできたプログラミング言語あるいはプログラミング環境は高校によって異なる

2. 授業内容と到達目標

私立大学情報教育協会で策定された「情報活用能力」において、この「アルゴリズムとプログラミング」は、問題解決という点で到達目標 A にも関連するが、到達目標 C の「情報通信技術の現状と可能性を考察し、論理的思考に基づき、価値創造に向けて必要となるIoT、モデル化、データサイエンス、AIなどの知識・技能を活用できる。」の一部として考える。この到達目標 C の到達点2-1「仮説検証の手段として、

1) 追手門学院大学経営学部, 2) 拓殖大学工学部, 3) 静岡産業大学経営学部

論理的思考に基づいてモデル化とシミュレーションなどを通じて予測することができる」には、次の3項目が挙げられている。

- ・ 現実の問題を体系的な観点で捉え、モデルを構築する手法を演習させる。
- ・ アルゴリズムを具体的なプログラムとして実現し、コンピュータで実行させる。ここでは、実用的なプログラミング技術の修得ではなく、問題解決のためのアルゴリズムを修得させる。
- ・ 構築したモデルからシミュレーションなどを用いて予測させる。

この2番目を中心とした授業内容案を作成することとし、次の5つの方針に沿って考えることとした。

1. 情報系だけでなく、すべての学部学科で用いられる教材とする。
2. PCを用いた演習とする。BYODかPC演習室かは問わない。
3. 教材は、Pythonをベースとして考える。もちろん、私情協で作成するのは一つの授業例であり、実際に使用する言語や環境は大学・学部・学科で自由に決めればよい。
4. 高校で使われるプログラミング言語は様々である(Python, JavaScript, VBA, C, Java, …)。したがって、大学での授業で用いる言語に慣れるために、最初の1~2回程度の授業ではゆっくり進める必要がある。
5. 1クラス30人程度で教員1名+TA(あるいはSA)2名程度で考える。実際のクラス運用については、受講者のスキルや授業規模(人数)、TA等の有無などに依存する。

3. 授業内容案

「アルゴリズムとプログラミング」について、以下のような90分×5回程度の授業で考える。情報処理入門的な2単位1科目の中に入れるか、独立した科目とするかは各大学の状況に依存する。

- 第1回:プログラミングの基礎:Pythonを用いてプログラミング方法の基本を復習する
- 第2回:配列データの探索アルゴリズム:線形探索・2分探索
- 第3回:関数と整列アルゴリズム:バブルソート・選択ソート・選択ソート、計算量の基礎
- 第4回:回転描画のアルゴリズム:タートルグラフィックスを用いた順次・繰り返し・条件分岐
- 第5回:再帰アルゴリズム:クイックソート(Computer Science Unpluggedのビデオも利用)などの簡単な再帰プログラムを扱う

4. シラバス例

回数	題目	授業内容	備考
1	プログラミングの基礎	Pythonによるプログラミング方法の基礎を習得する 1. プログラミング環境の準備 Google ColaboratoryとPythonの使い方 2. 変数と型 3. 逐次処理と繰り返し処理 4. 条件分岐 5. 関数、変数のスコープ	高校の新学習指導要領で必修化されたプログラミングの復習としての位置づけと、Pythonの導入である。学習者の受講前の状況に応じて補習が必要な場合も考えられる。

2	配列データの探索アルゴリズム	<p>配列の概念を学び、線形探索と二分探索のアルゴリズムの性質を理解する</p> <ol style="list-style-type: none"> 1. 配列の使い方を理解する 2. 代表的な線形探索と二分探索のアルゴリズムを理解する 3. プログラムを動作させて、データ比較処理の回数を計測し、探索対象となるデータの数や探索する値によって比較処理回数が異なることを理解する 4. 2つのアルゴリズムを比較して、効率の良いアルゴリズムを考える 	<p>高校で学習している可能性もあるが、計算量につなげて評価し、アルゴリズムの効率を考えられるようにする。なお、領域計算量と時間計算量のトレードオフについては扱わず、発展的内容とする。</p>
3	関数と整列アルゴリズム、計算量	<p>関数を導入し、整列のアルゴリズムを学ぶ</p> <ol style="list-style-type: none"> 1. 関数の使い方を理解する 2. 代表的な整列アルゴリズムとして、バブルソート、挿入ソート、選択ソートを理解する 3. プログラムを動作させ、比較操作と交換操作の回数を計測し、データの初期状態によって操作回数が異なることを理解する 4. 代表的なアルゴリズムの平均計算量と最悪計算量を O 記法で表現する 	<p>例えば、データ数が 1000, 2000, …, 5000 で、データが整列済みの場合について、比較操作回数を計測するプログラムの枠組み(スタブ)は与える。学習者に測定値をグラフ化させる。データが逆順の場合とランダムの場合も同様にする。</p>
4	図形描画のアルゴリズム	<p>タートルグラフィックスを用いた図形描画により、順次・繰り返し・条件分岐のアルゴリズムの理解度を深化させる</p> <ol style="list-style-type: none"> 1. タートルグラフィックスの基礎 <ul style="list-style-type: none"> ・正三角形、正方形、星形などの描画 ・図形の繰り返し描画 2. タートルグラフィックスの応用 <p>伝統的な連続文様の描画により、課題の要求や条件の分析、プログラムの分割、アルゴリズムの工夫などを理解する</p> <ul style="list-style-type: none"> ・市松模様、鱗文様、毘沙門亀甲文様、麻の葉文様などの描画 	<p>CUI (character user interface) 的な内容だけでなく、グラフィックスを扱うことで学習者の興味を引くことができる。順次処理、繰り返し処理、条件分岐の復習を兼ねる。</p>
5	再帰アルゴリズム	<p>コンピュータサイエンスの頻出概念としての再帰(recursion)を理解する</p> <ol style="list-style-type: none"> 1. 再帰処理を理解する <ul style="list-style-type: none"> ・1から n までの整数和 ・n の階乗など 	<p>高校の数学 B を履修しておらず、漸化式を知らない学生も多いので、教え方には留意する。 CSU(Computer</p>

		2. クイックソートなどを例に、再帰処理による問題解決を理解する 3. 再帰処理を利用した問題解決を考える 例えば、迷路作成、ハノイの塔など。	Science Unplugged)のビデオなども活用して概念を理解する
--	--	---	--------------------------------------

5. その他

1. 教科書は各実施担当教員が指定するものであり、私情協から教科書を出版あるいは提供するものではない。今回の教材の教科書として、オーム社の IT Text シリーズ「一般情報教育」(既刊)を考える。(上記の第 1 回、第 2 回、第 3 回、第 5 回)
2. PDF ベースのスライドあるいは説明書を検討する。担当者が自分の授業用に用意した資料を、私情協用にアレンジして提供することも検討している。(第 4 回の図形描画処理など)
3. 授業方法は伝統的な方法と反転授業が考えられる。上記のシラバス例はこの伝統的な授業の場合であり、対面形式で黒板や画面上のプログラムを説明して、演習させる。復習用に自習ビデオ教材を用意することも考えられる。

一方、反転授業では、ビデオ教材により基礎的な演習問題を実行させ、授業では演習問題の解説および応用問題の演習をする形式が考えられる。

ライブ(オンタイム)のオンライン講義では、受講者側環境はパソコン 1 台の場合が多い。このとき、受講者側では教員から送られてくる映像画面と、自分のプログラミング環境の画面を同時に見るのが難しいことに配慮しておく必要がある。受講者側で拡張ディスプレイを用意するなどできればよいが、現状ではそこまで期待できない。

4. プログラム(システム)の仕様記述、あるいは、プログラミングの作法(変数名の付け方、コメントの入れ方など)も学習してもらいたいが、100 行程度のプログラムを自分で作成するぐらいの経験をさせながら繰り返し指導する必要があり、この 5 回程度の内容に含めるのは難しい。

以上