

生徒による生徒のためのプログラミング講義 —数理科学部における取組—

愛知県立旭丘高等学校 田中 紀子

Tanaka Noriko, Aichi Prefecctural Asahigaoka High School

1 はじめに

高等学校に関わる教育の動向について述べる。

高等学校では、令和4年度から新学習指導要領に則った新教育課程が始まった。それに伴い、令和7年度からの大学入学共通テストは、数学では試験時間・試験範囲とも変更になり、また、新たに情報の共通テストがスタートする。すでにいくつかの大学が令和7年度の第2次学力試験の試験範囲を発表しており、東京大学は前期日程文科、前期日程理科ともに「統計的な推測」を出題範囲に加えた。別に、大阪大学は第2次学力試験で、文系・理系とも「統計的な推測」を範囲に加えなかった。

高等学校では、共通テストで「情報I」が受験科目に加わることや、数学B「統計的な推測」の扱いなどについて教員は気にかけており、学校によっては受験科目の内容が教育課程編成に影響を与えると思われる。

加えて、情報分野に関わるコンテストについて紹介したい。情報に関わるコンテストには、情報オリンピック、情報科学の達人、競技プログラミング等があり、また、総合的なコンテストのなかに情報やプログラミングの部門を含む大会として、SSH生徒研究発表会@神戸（主催：JST、SSH校対象：220校余）、日本学生科学賞（主催：読売新聞）、JSEC（高校生科学技術チャレンジ）（主催：朝日新聞）などがある。SSH生徒研究発表会を例にとると、近年、プログラミングを用いた生徒の発表の本数は増加傾向にあるといえる。

勤務校には数理科学部があり、部員たちは、数学や物理、化学、生物、地学、情報など、数名ずつでグループになってさまざまな探究活動を行っている。ここでは、情報オリンピックや情報科学の達人に参加した生徒が、下級生を対象に行った「生徒による生徒のためのプログラミング講義」を紹介したい。

2 研究内容

2.1 「生徒による生徒のためのプログラミング講義」

(1) 動機

講師をした生徒は情報オリンピックや情報科学の達人に参加し、面白いなど感じたことを、参加したことのない1年生や同級生にも伝えたい、広めたいと思ったことが、「プログラミング講義」を行うきっかけとなっている。プログラミングの面白さを伝えるため、興味深い部分を抽出して講義をしようと思ったと語っており、また、競技プログラミングで勝てるようにといったことが目的の一つになっている。

(2) 講義内容

講義は毎回パワーポイントに分かりやすくまとめて、授業後の部活動の時間帯に実施していた。以下、講義内容の一部（初回と第4回の一部）を抜粋して掲載する。（生徒の言葉をそのまま引用）

第0回 情報講義

○ 講義の前提

扱う言語：C++

理由

- 動作の速度が早いので、競技プログラミングに向いています
- JOI や AtCoder などの解説で C++ が使われています
- 一部のコンテストサイトではこれと JAVA しか使えない
- 色々便利
- いきなり C++ を学び始めるのは難しいので、第0回講義では、授業で扱った「十進ベーシック」で基本は解説します

注意事項

- 今回の講義では、簡単のため厳密さや正確さを欠いている部分が多々あります
- しかし、それらを追求してしまうと、非常に難解な説明かつ、長い時間が必要になります
- よって、少々簡単な説明に留めます。

○ 前提知識を復習

- 変数
 - 数字や文字を入れられる
 - だいたいなんでも入れられる
 - 10進ベーシックでは LET で宣言
- 関数

- コンピュータに動作をさせるプログラム
- print: 出力する sort: 並べ替える max:最大値を取るなど多種多様
- 自分で定義をすることも可能
- 演算子
 - 数字と数字を一定のルールの上で計算するもの
 - 以下のようなものがあります
 - 演算系
 - + - : 足し算 引き算を行う $3 + 2 \rightarrow 5$
 - * / : 掛け算 割り算を行う $3 * 2 \rightarrow 6$
 - 比較系 true false を出力する
 - > < : 大小比較をする $3 > 5 \rightarrow \text{false}$
 - == : 等しいか比較する $4 == 8/2 \rightarrow \text{true}$
- if 文
- for 文
- 素数判定プログラムを作成！
 - (10進ベーシックで作成)
 - 必要なもの
 - INPUT 関数 “INPUT 変数名” 変数にキーボードから入力
 - PRINT 関数 “PRINT 変数名” 変数の中身を出力をすることができる
 - SQRT 関数 “SQRT(a)” a の平方根になる
 - MOD 関数 “MOD(a,b)” a を b で割ったあまりになる
 - 制約
 - 入力する値は、 $2 < \text{入力} < 1024$ の整数とします

< 2つの解答例を示して説明 >

- このように、数学的な工夫やプログラムの工夫によって、プログラムの計算量を少なくしたり、プログラムをより簡単に書けるようにすることが求められる

○ 予備知識の学習

- 配列
 - 変数の列と考えることができる
 - 配列の要素にアクセスしたい場合は、配列名(場所)と書くことで普通の変数として扱えます。
 - 0 から index が始まることに注意
 - 10進ベーシックで扱うときは先頭に”OPTION BASE 0”というおまじないが必要
- 計算量とは
 - そのプログラムが計算にどれだけ時間がかかるかの見積もり

- アルゴリズムの A の計算量 $T(N)$ が概ね $P(N)$ に比例することを $TN=O(PN)$ であると表し、アルゴリズム A の計算量を $O(PN)$ であるという (N は入力サイズ) (計算量とは何回プログラムを実行するかということ)

< 素数判定プログラムで具体例を考える >

○実際に問題に触れてみよう!

- ここからは実際にプログラミングの面白さに触れていこうと思います。
- プログラムを組むことは流石に難しいので、紙に指針を書いてみてください。
- 初学者でも考えつきそうな問題を 2 つ用意しました。

問 1 長さ L cm の竿の上を N 匹のアリが毎秒 1cm で動いています。アリが竿の端にたどりつくると竿から落ちます。竿は横幅が狭いのですれ違うことができず、二匹のアリが出会うと、お互い U ターンして反対の方向に進み始めます。各アリについて、竿の左端からの現在の距離 X_i はわかりますが、向いている向きはわかりません。すべてのアリが落ちるまでにかかる最小の時間と最大の時間を求めてください。

- 制約 $L: 1$ 以上 10^6 以下, $N: 1$ 以上 10^6 以下, $X_i: 0$ 以上 L 以下
- 計算回数の上限は 10^9 乗とします

問 2 年齢当てゲーム

目の前に 0 から 127 歳の人間がいます。その人に Yes/No で答えられる質問をして、年齢を確定させたいです。最低でも質問は何回必要でしょうか?

以上で今回の講義は終了になります

もし面白いと思ったら次回以降も来ていただけると嬉しいです

第 4 回 情報講義

○グラフとは?

- 頂点と辺を持った構造
- 物事の関係を表すのに用いる。
- グラフで表せるものの例
 - SNS の FF 関係
 - 物の輸送経路
 - 人間の交友関係
 - and more !
- 大きく分けて 2 種類のグラフがあります。
 1. 有向グラフ
 - 辺が向きを持つるグラフで、矢印の向きにしか進めません。
 2. 無向グラフ
 - 向きを持っていないグラフで、両方に進めます。

- プログラミングで実現する場合
- 各頂点に 0-indexed の番号をふる
- 二次元配列を作成

○ DFS

- 最も有名かつ初歩的なグラフ探索手法です。
- グラフ探索手法とは、グラフから何かしらの情報を読み取ることの総称です。
- DFS では、パスが存在するかどうかを調べることができます。
- パスとは、ある頂点からある頂点への道のりです。

● DFS とは

- Deep First Search の略で、日本語だと深さ優先探索
- ある頂点 s を始点として、すべての頂点 t に対して $s-t$ パスが存在するかを判別することができる。
 - どのように探索するか
 - 始点から出ている辺の先の頂点を調べる
 - その頂点から出ている辺の先の頂点を調べる
 - …以下ループ
 - すべての頂点の探索が終わり次第終了

<例を提示して説明>

○ BFS

- Breadth First Search の略
 - これを用いることによって、頂点からの距離がわかる。
- どのように実装するか
 - 辺をデータ構造の queue に入れて、頂点から出る辺を探索する
- データ構造 queue とは？
 - FILO First In Last Out 型のデータ構造
 - 人が並んでいる列を考えると良い

<例を提示して説明>

○ DFS と BFS 何が違うの？

- DFS は、FILO First in Last out
- BFS は、FIFO First in First out
- DFS はつきすすむ探索手順
- BFS はまんべんなく進む探索手順
- 計算量はともに $O(E+V)$ です。(最悪の場合でも、頂点を 1 回、辺を 1 回を全部確認するため)
 - It ' s so Fast !

2.2 担当生徒のコメント

- プログラミングの敷居を下げる努力をする必要がある。
- 学習コスト（時間＋労力）が高い。一から学ぶ必要があり、学校が教育すべき。
- 説明するには勉強が必要。コンテストで勝つためには、ひらめき、引き出し多さ、発想力の豊かさが必要。
- 10進ベーシックは、携帯電話に例えるとガラケー的で、古すぎる、機能もよくなり、きっちりとした書き方になっていない。
- 役立つのはC言語、C++で、数学に例えると大学数学（理論がきちんとしている）。
- Pythonはブラックボックスになっている（内部で処理している）。できあがっているコンテンツが多い。
- 処理速度は、10進ベーシック < Python・Java < C, C++

3 考察

「生徒による生徒のためのプログラミング講義」を通じて、生徒同士が数学ソフトウェア、プログラミングで学び合うために必要なことについて述べておく。

一つには、生徒たちが自由に学び、教え合うための環境の提供（教室や時間の確保）が必要であろう。講師をした生徒は、毎回パワーポイントに分かりやすくまとめ、丁寧に教えていた。プログラミングが好きで、自らも積極的に学ぶとともに、それを伝えたいという熱意のある生徒がいなくてこういう活動は続かない。

さらに、聴講する側にも「学び」が必要で、聴講生徒にも熱意がないと続かない。

このような発展的な学びの場合、教師は指導ではなく支援するという立場であることが適当で、ファシリテーターとしての役目を果たすほうがよいと考えている。

教える学びから探究的な学びへと、学びの形態が変化するとき、教師にとって大切なことは、教師自身が学び続けることであろう。

参考文献

- [1] 情報オリンピック日本委員会

<https://www.ioi-jp.org/>(R 4年 11月 24日取得)

- [2] 高校生科学技術チャレンジ (JSEC)

<https://manabu.asahi.com/jsec/>(R 4年 11月 24日取得)

- [3] 国立情報学研究所グローバルサイエンスキャンパス 情報科学の達人
<https://www.nii.ac.jp/tatsujin/>(R 4年 11月 24日取得)
- [4] 国立研究開発法人 科学技術振興機構 次世代人材育成事業 スーパーサイエンスハイスクール
<https://www.jst.go.jp/cpse/ssh/index.html>(R 4年 11月 24日取得)
- [5] 文部科学省：高等学校学習指導要領（平成30年告示）解説【数学編 理数編】，2018.
- [6] 日本学生科学賞
<https://event.yomiuri.co.jp/jssa/>(R 4年 11月 24日取得)