

問題解決から導くプログラミング教育方法の案

和田 勉
長野大学 産業情報学科
ben@nagano.ac.jp

情報処理学会情報処理教育委員会・初等中等情報教育委員会での、高等学校に新設される「情報」教科のあるべき内容を示すことを目的としたモデル教科書執筆プロジェクトの一部として、情報B(3)「情報処理の定式化とデータ管理」の後半の執筆を行なった。ここでは章の前半の部分での陸上競技大会を例にした問題の発見・分析等を受けて、その一部を「各競技へのコース割り当て問題」として取り出し、これの解法を探る過程でアルゴリズムやプログラミング的な考え方へ導くことをねらった。本稿ではこの教科書案を紹介し、これの試験の利用を踏まえての、この種の教えかたの長所と欠点、適切な用い方に関する考えを述べる。

A proposal of an education method for programming - leading students in solving problems

Ben Tsutom WADA
Division of Industry and Information Science
Nagano University
ben@nagano.ac.jp

As a part of an IPSJ committee's project for writing a textbook on the subject "Information" which will be established as a new subject in senior high schools, a project for the sake of showing its ideal contents, I wrote the second half of the section Information B(3) "Formalization of information processing, and data management." Corresponding to the first half of the section, in which problem finding, problem analyzing, etc. are discussed along the example "track athletic events," this second half extracts a part of it, "allocation problem of tracks to each game," and seeks its solution, aiming to introduce students to the viewpoint of algorithms and programming. In this paper, I introduce this draft, and discuss on the advantages and drawbacks of this kind of teaching method, and on its appropriate usage.

0. 序論

情報処理学会情報処理教育委員会・初等中等情報教育委員会での、高等学校に新設される「情報」教科のあるべき内容を示すことを目的

としたモデル教科書執筆プロジェクトの一部として、執筆の一部の担当が私にも割り当てられ執筆したことはすでに報告した。[和田98]しかしその後、教育課程審議会の答申[教育

課程審 98]などが出されたことや、プロジェクトチーム内部での討論の状況から、あらためて章建てを大きく組み替えて再度モデル教科書を執筆することになり、私は新たに情報B(3)「情報処理の定式化とデータ管理」の後半の執筆を受け持つことになった。これはこの章の前半の部分で、陸上競技大会を例にした問題の発見・分析・抽出を行っている[神沼 98]のを受けて、その一部を「陸上競技大会の各競技へのコース割り当て問題」として取り出し、これの解法を探る過程で、アルゴリズムの考え方、さらには(プログラミング言語は用いないが)プログラミング的考え方へ導くことをねらったものである。

以下にこの私の執筆部分を示し、続いてこれを用いて行った試験的授業の結果と、それに基づく考察を述べる。

1. 陸上競技割当問題

設定問題：

この競技大会を行なう会場として、図のような400メートルのコースが使えることになった。

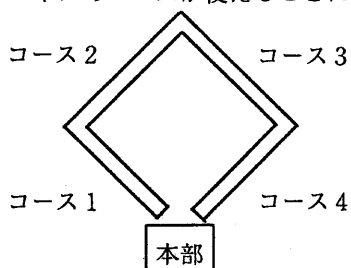


図1. 競技コース

図のコース1～4はそれぞれ長さは100mの直線である。100 m走、100 mハードル走はコース1～4のどこでも行なえる。200 m走、200 mハードル走はコース1とコース2、コース2とコース3などのように、コース2つをつなげて行なう。ただしコース1とコース4

//陸上競技のコース割り当て 方法1

// コース1～4の4つのコースに各競技を割り当てる

//

コース1で100 mの競技全部を順次行なう。

コース2～4をつなげて200 m、300 mの競技全部を順次行なう。

は間に競技本部が設置されるので、つなげて使うことはできない。300 m走も同様に、例えばコース2、3、4のように3つのコースを使って行なうとする。

ここで以下の競技を行なうことを考える。

100 m走A

100 m走B

200 m走

300 m走

100 mハードル走

200 mハードル走

それぞれにかかる時間は次のように予定されている。

・100 m・200 m・300 m走はそれぞれ30分

・100 m・200 mハードル走は準備・後始末に

時間がかかるのでそれぞれ45分と60分

これをまとめると下のようになる。

100 m走A 30分

100 m走B 30分

200 m走 30分

300 m走 30分

100 mハードル走 45分

200 mハードル走 60分

それぞれの競技をどのコースでどのようなスケジュールで行なったらいいだろうか。

演習：それぞれの競技をどのコースでどのようなスケジュールで行なったらいいか考えてプランを作ってみよ。またそのスケジュールでは始めてから終るまで時間がどれだけ必要か考えてみよ。

いろいろなものが考えられるが、例えば方法1のようにすると、全体が終るのに2時間かかる。

この競技場は使う時間が長いとそのぶん利用料を多く支払わなければならないので、始めてから終るまでの時間はできるだけ短くしたい。競技の割当を工夫して、終る時間をもっと早くすることを考えてみよう。

//陸上競技のコース割り当て 方法2-第1段階

// コース1~4の4つのコースに各競技を割り当てる

//

コースが空いたら

空いたコースで行なうことのできる中で一番距離の長い競技を行なう

(同じ距離の競技がいくつもある場合は

その中で時間が最もかかる競技を行なう)

ここで以下の方法に基づいて割り当てることを考えよう。(方法2-第1段階:上図)

この説明を、「実際にコースを割り当ててスケジュールを作る方法」の形に書きなおしてみよう。割り当ては、大会開始の時刻から始めて時間を順に追って行なうが必要になる。すなわち

1.まず大会開始時点の割り当てをする。(コースは最初は全て空いているのでそこに上の方法に基づいて競技を割り当てる。)

2.その後は、開始から15分後の状況、30分後の状況、と、15分ごとの時刻を順次想定して状況を考えてゆく。それぞれの時刻において、その時終わった競技がありコースが空いたら、上の方法に従って別の競技を割り当てる。

3.全ての競技が終わったら大会を終了すると進めていくことになる。これを考慮して手順を書いてみよう。(第2段階)

このように時間を順に追って割り当てを進

めるためには「いま開始何分後の作業をやっているのか」を控えておくが必要になる。すなわち紙の上などに「現在時刻」という場所を設けて

- ・最初にそこに0分と書く
- ・大会開始時点の競技割当を行う
- ・0分を15分に直す
- ・開始後15分の状況での割当作業を行う

.....

のように進めていくことになる。

また、上記の2で、一つ競技が終わってコースが空いても開始できる競技が無く、コースをしばらく空いたままにせざるをえなかったり、またその後で再度そのコースで競技を開始することになる等の場合もある。そのため「このコースはいま空いている」というしるしをつけておく必要があるため、そのための場所も用意しておく。

//陸上競技のコース割り当て 方法2-第2段階

// コース1~4の4つのコースに各競技を割り当てる

//

すべての競技が終了するまで時刻を15分づつ進めながら繰り返す:

この時刻に競技が終了したコースがあれば

そのコースのしるしを「空」に変える

(終了したコースがいくつもある場合は上記を繰り返す)

空いた/空いているコースがあれば

実行できる中で最も距離の長い競技を選んで開始する

(距離が同じ競技がいくつもある場合は最もかかる時間の長い競技を選ぶ)

(競技を開始できた場合は上記を繰り返す)

—もう一つ同時に開始できる場合もあるので

繰り返しはここまで

演習：「一つ競技が終わってコースが空いても開始できる競技が無く、コースをしばらく空いたままにせざるをえない」のはどんな場合があるか。

現在時刻

60分

これらの考え方を使って、この方法をより正確に書き直してみよう。(第3段階)

コース 1	使用中
コース 2	使用中
コース 3	空
コース 4	使用中

//陸上競技のコース割り当て 方法2-第3段階
// コース1~4の4つのコースに各競技を割り当てる
//
繰り返す:

//この時刻に終了した競技があればそのコースのしるしを「空」に変える
コース1~4それぞれについて:

そのコースでその時競技が終了するなら
そのコースのしるしを「空」に変える

//コースの印を「空」に変える処理はここまで

//空いているコースがあれば割り当てる-可能な限り繰り返す
繰り返す:

//空いている最長コースを調べる

//単独の空いているコースがあるかを調べる

コース1~4それぞれについて:

そのコースが空ならば

コースが1つ空いていることと、コース番号を控えておく

//2つつなげて使えるコースがあるかを調べる

コース「1と2」「2と3」「3と4」それぞれの組について:

その両方のコースが空ならば

コースが2つ空いていることと、コース番号2つを控えておく

//3つつなげて使えるコースがあるかを調べる

コース「1と2と3」「2と3と4」それぞれの組について:

その3つのコースが空ならば

コースが3つ空いていることと、コース番号3つを控えておく

//空いている最長コースを調べる処理はここまで

//空いているコースがあり、割り当てられる競技があれば割り当てる
空いた/空いているコースがあるならば

今空いているコースの長さで実行できる最も長距離の競技を選ぶ

そのような競技があるならば

そのような競技(最も長距離の競技)が2つ以上あるならば

その中で必要時間が最大のものを選ぶ

//割り当てを行なう

空いているコース中の、競技に使うコースにだけ:

使用中の印を付ける

競技が終わりコースが空く時刻を控えておく

//割当の処理はここまで

//空いているコースへの割り当て処理はここまで
繰り返すはここまで一競技を割り当てたならもう一度繰り返す
// さらにまた別の競技も始められるかも知れないので繰り返し試みる
「現在時刻」を15分進める
繰り返すはここまで一コースが一つでも使用中なら繰り返しを続ける

演習：上記の方法を忠実になぞるとどのように競技が割り当てられていくかやってみよ。
始めてから終るまで時間は何分かかったか。

このようにあること（この場合は競技へのコースの割り当て）を行なう方法をきちんと
書いたものをアルゴリズムという。

演習：上記の方法2も必ずしも最良の割当をするものではなく、行なう競技の個数・距離・時間
を変えると無駄なコースの割り当てを行なうこともある。より無駄の少ない割当をするような
別のアルゴリズムを考えてみよ。それによるコース割り当てをいろいろな場合について方法2で
のものと比較してみよ。

上記のアルゴリズムはまだ細かい部分を省略して書いてある。同じアルゴリズムをより細かく
書いたものを下に示す。（第4段階）

//陸上競技のコース割り当て アルゴリズム（方法）2—第4段階
// コース1～4の4つのコースに各競技を割り当てる
//
//記憶場所を用意する
各競技種目の名前・必要距離・時間を覚えておく場所「競技」を用意し、
各競技種目の名前・必要距離・時間を設定する
各コースが「使用中」か「空」か、および終了時刻を覚えておく場所
「コース1」から「コース4」を用意し、最初は「空」にする
現在時刻を覚えておく場所「現在時刻」を用意し、最初は「0分」にする

//割り当てを行なう
繰り返す：

//この時刻に終了した競技があればそのコースのしるしを「空」に変える
iを1～4まで変えながら：
もし コースiが使用中で
かつ 現在時刻=コースiの終了時刻 ならば
コースiを空にする
//コースの印を「空」に変える処理はここまで

//空いているコースがあれば割り当てる—可能な限り繰り返す
繰り返す：

//空いている最長コースを調べる
場所「最大コース長」（最初は0とする）と「候補コース」を用意する
//単独の空いているコースがあるかを調べる
jを1～4まで変えながら：
もし コースjが空 ならば
最大コース長を1に、候補コースをjにする
//2つつけて使えるコースがあるかを調べる
もし（コース3が空 かつ コース4が空）ならば
最大コース長を2に、候補コースを{3,4}にする

もし (コース2が空 かつ コース3が空) ならば
最大コース長を2に、候補コースを[2,3]にする
もし (コース1が空 かつ コース2が空) ならば
最大コース長を2に、候補コースを[1,2]にする
//3つつけて使えるコースがあるかを調べる
もし (コース2が空 かつ コース3が空 かつ コース4が空) ならば
最大コース長を3に、候補コースを[2,3,4]にする
もし (コース1が空 かつ コース2が空 かつ コース3が空) ならば
最大コース長を3に、候補コースを[1,2,3]にする
//空いている最長コースを調べる処理はここまで

「開始競技」を「なし」に設定する

//空いているコースがあり、割り当てられる競技があれば割り当てる
もし 最大コース長 > 0 ならば
「競技」の中の、必要距離が最大コース長以下のものの中で
最も長いものを選び「開始競技」とする
もし 「開始競技」がある ならば
もし 「開始競技」が2つ以上ある ならば
その中で必要時間が最大のものだけ「開始競技」とする
//割り当てを行なう
「開始競技の必要距離」の回数繰り返す：
//距離100mの競技なら1回、200mの競技なら2回繰り返す
「候補コース」の先頭のコースを使用中にする
「候補コース」の先頭のコースの終了時刻を
現在時刻+「開始競技の終了時刻」に設定する
「候補コース」から先頭のコースを削除する
「競技」から開始競技を削除する
//割当の処理はここまで
//空いているコースがあっても、そこでできる
// 開始競技がない場合は割り当てはできない
//空いているコースへの割り当て処理はここまで

繰り返しはここまで「開始競技」が「なし」でないなら繰り返しを続ける
// さらにまた別の競技も始められるかも知れないので繰り返し試みる

「現在時刻」を15分進める
繰り返しはここまでコースが一つでも使用中なら繰り返しを続ける

アルゴリズムをここまで詳しく正確に書けば、コンピュータもこれを理解できるようになる。
ただしコンピュータに理解できる、「プログラミング言語」という決まった形式に従って書かなければならない。これをプログラムという。これをコンピュータに与えると、上の演習で手作業でやったのと全く同じことをコンピュータは忠実に超高速で行なってくれる。

上のアルゴリズムをプログラミング言語で書いたものを以下に示す。プログラミング言語には多くの種類がある。ここではC++ (シープラスプラス) 言語でのものを示す。

```

//陸上競技のコース割り当て アルゴリズム (方法) 2 c++言語によるプログラム
// コース1~4の4つのコースに各競技を割り当てる
void main()
{
    //記憶場所を用意する
    t_kyoudi kyoudi; //各競技種目の名前・必要距離・時間を覚えておく場所
    //各競技種目の名前・必要距離・時間の設定
    kyoudi.settei("100 m走A", 1, 30);
    kyoudi.settei("100 m走B", 1, 30);
    kyoudi.settei("200 m走", 2, 30);
    kyoudi.settei("300 m走", 3, 30);
    kyoudi.settei("100 mハードル走", 1, 45);
    kyoudi.settei("200 mハードル走", 2, 60);

    //各コースが「使用中」か「空」か、および終了時刻を覚えておく場所
    t_course course;
    //最初は「空」にする
    course.shiyou[1] = aki;
    course.shiyou[2] = aki;
    course.shiyou[3] = aki;
    course.shiyou[4] = aki;

    //現在時刻を覚えておく場所
    int genzaijikoku = 0; //最初は「0分」にする

    //割り当てを行う
    do {
        //この時刻に終了した競技があればそのコースのしるしを「空」に変える
        for (int i = 1; i &lt;= 4; ++i) {
            if (course.shiyou[i] == shiyoochuu
                && genzaijikoku == course.shuryoujikoku[i]) {
                course.shiyou[i] = aki;
            }
        }
        //コースの印を「空」に変える処理はここまで
        .....
    }
}
(紙数の都合でプログラムの以降の部分は省略)

```

コンピュータが動くには必ずプログラムが必要である。どんな時もコンピュータが動いている時には、その中にはかならず、誰かが作った上の例のようなプログラムが入っており、コンピュータはそれに忠実に従ったことを超高速に行なっているにすぎないのである。

2. 試験的授業とその結果の考察

上記の教科書案を使い、本学産業情報学科1年生(数理的なことに強い興味を持つ者は少ない)対象の授業で、試験的にこれに沿った解説を試みた。結果は惨澹たるもので、30人ほどのクラスがほとんど全員お手上げという状況で、このクラス中では比較的優秀な者すら「自分などにはこういうことはとてもわからない」と投げつけてしまうありさまであった。

この題材を選ぶに際しては、現実の問題の発見・分析に端を発する「陸上競技大会」(あるいは他の何らかのイベント)の運営で生じる種々の問題の中から題材を選ぶ、ということであらかじめ決めたほか、プロジェクトチーム内での打ち合わせに基づき、いくつかのことに留意して選んだ。その最も大きなものは、「特定のアルゴリズムに基づかずに直感的に漠然と考えていては解がわからない、あるいは間違え

やすい」問題を選ぶ、というものである。これは、アルゴリズム的思考の優位さを示すねらいであった。

もちろん高度で複雑な問題ならばそのようなものはいくらでもあるが、それでは教科書の題材としては不適當である。一方、単純な問題の多くは直感的に考えても最適解が得られてしまう。イベント運営で生じる問題で、手頃な規模と複雑さで、なおかつアルゴリズムに基づかずには漠然と考えていたのでは良い解に到達しにくいもの、という困難な設定を自らに課した中で、どうかその条件をある程度満たすと考えて選んだのがこの課題と解法である。

しかしそれでもこれだけ複雑(初心者にとっては)なものになってしまった。現実社会・現実の生活の問題から題材をとる「トップダウン」の教育方法は、動機づけの点からは良いことなのだが、この問題から逃れがたいという欠点がある。一方[和田 98]で述べたタートルグラフィックスのような、人工的なマイクロワールドから出発する「ボトムアップ」の教育方法はその裏返しの長所と欠点を持つ。すなわち初心者に十分把握できる単純な問題から出発することが容易で、その発達にあわせて徐々に複雑さを上げてゆくことができる。反面、学習者に対する動機づけは「幾何学的な模様が描かれるのが興味深い」などのたぐいの人工的な動機づけに頼らざるをえない。これは現実に基づかない、すなわち「底が浅い」もののため、強い動機づけを長続きさせることが困難である。また学習者により同期づけられる強さが大きく異なる。(「きれいな模様が描けていったいそれが何の役に立つのだ」などの反応をする学習者もいる。)

私は現時点では：

- ・最初にたとえ「お手上げ」であってもトップダウン的導入をとにかく紹介して学習者の記憶に残しておく、
- ・次いでしばらく、人工的動機づけに頼りながらボトムアップ的方法により学習させて可能な限り学習者の能力を向上させ、

・最後に元のトップダウン的導入で用いた題材に戻ってその数理解得、それとともに、それまでボトムアップで学んできたことも現実的・社会的にも意味があることを認識させる。

という「掘り下げ&昇り迎え」が一つの良い方法であろうと感じている。

なお、ここで述べた問題を「最適なコース割り当ての問題」と変えると、NP-complete になってしまう。これを避けるため、話の流れが「最適解を求める」すなわち「全ての組み合わせを調べてみて比べる」方向に行かないように配慮した。しかしそれでも、これがもし実際に高校の教科書になった場合、生徒や、特に高校の先生方が、「全ての組み合わせを調べてみて比べる」ことを考え付き、それでよしとしてしまう(計算量の問題には気が付かずに)懸念は指摘されている。

謝辞

共に活動している、このプロジェクト sigps のメンバ、とりわけ指導的立場である大岩元先生(慶応義塾大学)、武井恵雄先生(帝京大学)、私の上流の担当として多大なヒントをいただいた神沼靖子先生(前橋工科大学)、貴重な御意見をいただいたメンバ諸氏に感謝する。

参考文献

[和田 98] 和田勉「高等学校「情報」科目の教科書案—情報B(2)「プログラムとは何か」および情報C(2)」情報処理学会コンピュータと教育研究会 98-CE-49 平成10年7月31日

[教育課程審 98] 文部省教育課程審議会「幼稚園、小学校、中学校、高等学校、盲学校、聾学校及び養護学校の教育課程の基準の改善について」平成10年7月29日

[神沼 98] 神沼靖子「問題発見から解決過程の実践教育」平成10年度情報処理教育研究集会(文部省・九州工業大学)平成10年10月30日