第14回全国高等学校情報教育研究会全国大会（大阪/オンライン）
2021年8月10/11日

大学入試を見据えた教えないプログラミング教育
応用力の育成を考慮したプログラミング教育

普通の高校生に、どうやって配列やソートを教えるの???

大学入試のプログラミングやシミュレーションにどうやって対応するの

パワポ100%+ Amazon Poly 音声合成

関東第一高等学校
千葉県立生浜高等学校
情報科非常勤講師 太田 剛

説明内容

1. はじめに:目的と大学入試
2. プログラミング教育の現状と難しさ
3. プログラミングの認知過程を考える
4. 2020年度の教材と授業方法
5. 大学入試に向けた今後の計画

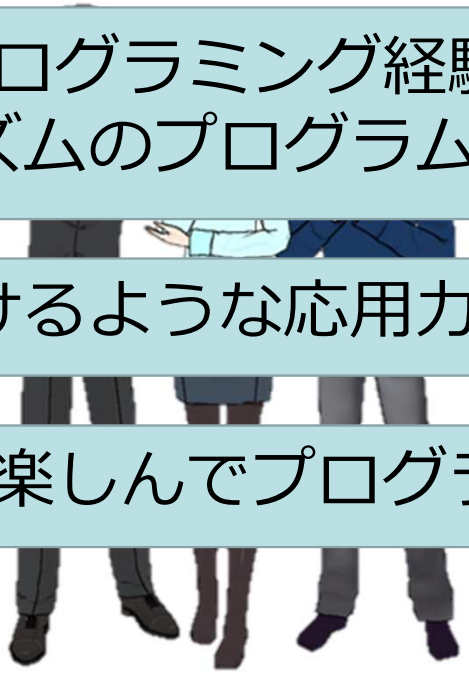
1 はじめに:目的と大学入試

1 はじめに:目的と大学入試

プログラミングが得意でない先生でも授業ができる。

年間シラバスから6時間程度の授業とする。

普通の学校の生徒で、プログラミング経験が無くても並び替えぐらいのアルゴリズムのプログラムを作成できる。



大学入試問題が解けるような応用力の基礎を作る。

生徒が主体的、対話的に楽しんでプログラミングを学習する。

一つのゴールとして大学入試

基礎的な問題	
基本	関数(再帰含む), 乱数の利用、リスト・配列処理、WebAPIの利用
アプリケーション機能	簡単な統計機能，時刻や日時の計算機能，文字列の処理機能
平面での物体の操作	マス目上でのロボットの移動制御，図形の描画
パズル問題	ハノイの塔やFizzBuzz，マス目パズル等
数学問題	一次方程式，素数や因数分解等

の
試
関

応用的な問題	
情報数学問題	パリティ検査,ハミング符号,ガロア体とフェルマーの小定理
ロボット制御	センサーからのフィードバックがあるロボットの制御
最適化/リソース割当	部屋割つけ，配車計画等
物理現象	振り子の動作，動体の移動等
グラフ/最短経路	最短経路: 移動距離等
予想/平均変化率	売り上げ予想, ローン計算等
待ち行列	ファーストフード店，交通渋滞
ライフゲーム	伝播状態: 感染状態等
確率関係	モンテカルロ法，ベイズ統計等

プログラミング又はシミュレーションの中問題、大問題として出題されることが多い

具体的な入試問題(1)

大学入試センター H30

第 3 問　（選択問題）　次の文章を読み，下の問い（問 1 ～ 3 ）に答えよ。（配点　35）

春夫君の高校では，授業で作った俳句の品評会を行い，表彰することとなった。品評会では 1 年生から 3 年生までの全校生徒による投票を行い，得票数の多い順に順位をつける。ただし，同じ得票数の場合は，先に提出した生徒の方を上位とし，同時提出は考えないものとする。この順位を使って次の規則で表彰する生徒を選ぶ。

> **金賞**　全体の中で順位が 1 位となる生徒 1 名を金賞に選ぶ。
>
> **銀賞**　金賞に選ばれた生徒の学年以外の 2 学年から，その学年の最高順位となる生徒 1 名を銀賞に選ぶ。したがって，銀賞の受賞者数は 2 名となる。
>
> **銅賞**　金賞および銀賞に選ばれていない生徒の中で，順位の高い生徒 5 名を銅賞に選ぶ。ただし，受賞者が一部の学年に偏りすぎないように各学年の銅賞受賞者が 3 名以下となるようにする。

春夫君は，投票の結果が記された名簿をもとに表彰する生徒を選ぶ手続きを作ることにした。なお，各学年から十分な数の俳句が提出されており，すべての賞は必ず人数分選ばれるものとする。

```
(21)  y ← 金賞に選んだ生徒の受付番号
(22)  dosyosu ← 0
(23)  繰り返し，
(24)      saidai ← -1
(25)      i を 1 から ninzu まで 1 ずつ増やしながら，
(26)          もし［ ス ］または（［ セ ］かつ y < i）ならば
(27)              もし［ カ ］< ［ キ ］ならば
(28)                  ［ ク ］← ［ ケ ］，x ← ［ コ ］
(29)              を実行する
(30)          を実行する
(31)      を繰り返す
(32)      g ← Gakunen[x]
(33)      もし Zyusyo[x] ＝「なし」かつ［ ソ ］ならば
(34)          Zyusyo[x] ←「銅賞」
(35)          Namae[x] と「を銅賞に選ぶ」を表示する
(36)          SyoNinzu[g] ← SyoNinzu[g] +1
(37)          dosyosu ← dosyosu+1
(38)      を実行する
(39)      y ← x
(40)  を，dosyosu ＝ 5 になるまで実行する
```

図 3　銅賞の生徒を選ぶ手続き

具体的な入試問題(2)

大学入試センター H31

第 3 問 （選択問題） 次の文章を読み，下の問い（問 1 ～ 3）に答えよ。（配点 35）

吉野さんはロボットを操作して宝探しをするゲームを作成している。図 1 はゲーム画面の完成予想図である。ゲーム画面には，横 YOKO マス×縦 TATE マスで構成されたゲームボード（以降，ボードと呼ぶ），各種情報，ロボットを操作するためのボタンが表示される。ボードのマスには宝が一つ，罠（わな）が複数隠されている。ゲームの目的は，ボード上のロボットを上下左右に 1 歩（＝ 1 マス）ずつ移動させ，罠を探知して避けながら，決められた操作回数以内で宝のマスに入れることである。

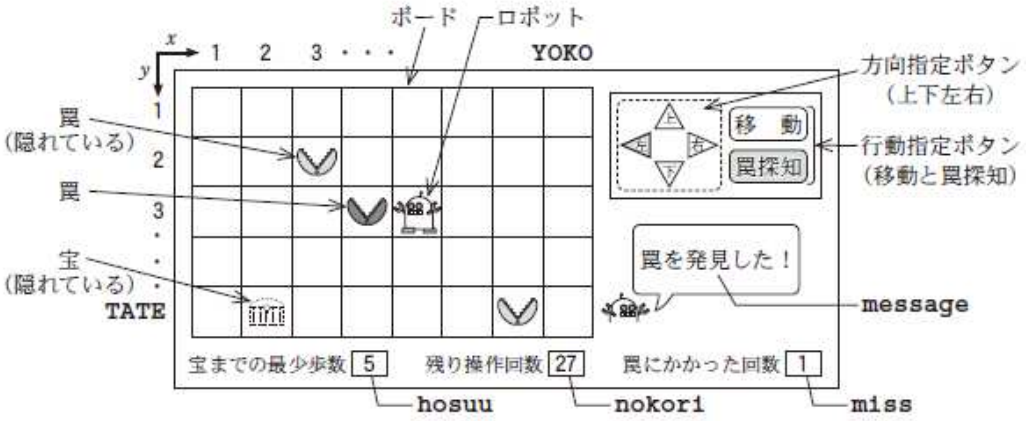


図 1 ゲーム画面

```
(01)  もし robo_x + d_x > 0 かつ robo_x + d_x ≦ [ウ] かつ
          robo_y + d_y > 0 かつ robo_y + d_y ≦ [エ] ならば
(02)  |   robo_x ← robo_x + d_x
(03)  |   robo_y ← robo_y + d_y
(04)  を実行する
(05)  nokori ← nokori − 1
(06)  もし takara_x = robo_x かつ takara_y = robo_y ならば
(07)  |   message ←「宝を見つけた！　宝探し成功！」
(08)  を実行する
```

図 2 移動ボタンが押されたときの手続き

```
(01)      （指定した方向に対応する値を d_x，d_y に代入）
(02)      もし押されたボタンが移動ボタンならば
(03-10)   |  （移動ボタンが押されたときの手続き（図 2）と同じ）
(11)      を実行し，そうでなくもし押されたボタンが
          罠探知ボタンならば
(12)      |  i を 1 から [キ] まで 1 ずつ増やしながら，
(13)      |  |  もし Wana_x[i] = robo_x + d_x かつ
          |  |      Wana_y[i] = robo_y + d_y ならば
(14)      |  |  |  message ←「罠を発見した！」
(15)      |  |  |  [コ]
(16)      |  |  を実行する
(17)      |  を繰り返す
(18)      |  [サ]
(19)      を実行する
```

図 5 行動指定ボタンが押されたときの「移動」と「罠探知」の手続き

情報の大学入試問題の怖さ

基礎的な問題
基本
アプリケーション機能
平面での物体の操作
パズル問題
数学問題

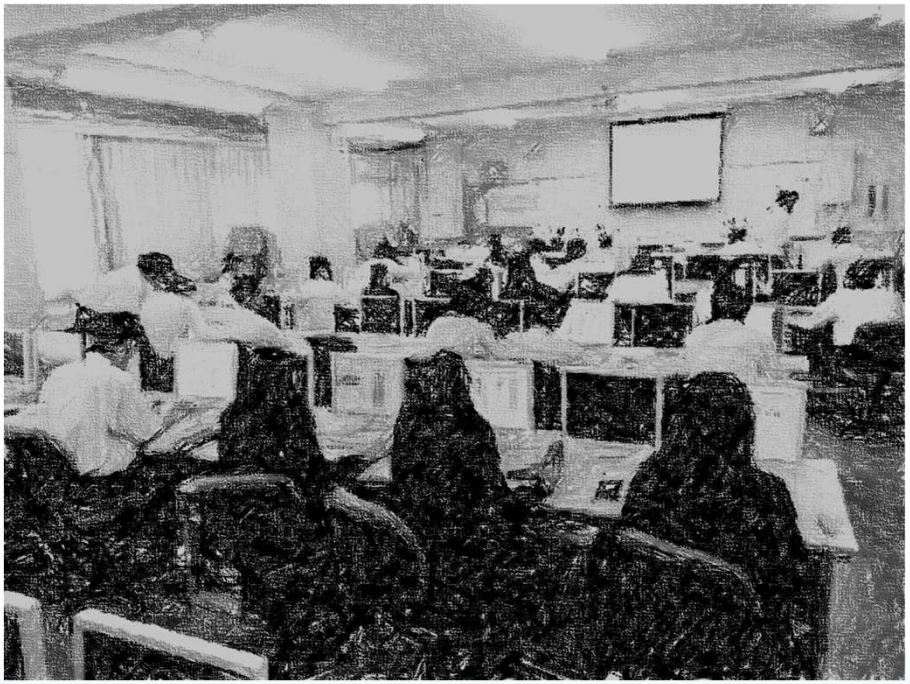
応用的な問題
情報数学問題
ロボット制御
最適化/リソース割当
物理現象
グラフ/最短経路
予想/平均変化率
待ち行列
ライフゲーム
確率関係

大学入試問題
・応用問題の考え方を使いながら、姿を変えていろいろ出題ができる。 ・極体に言うとアルゴリズムで表現できるならば、どんな難しい状況についても出題できる。

・応用問題(例題)の解き方を暗記するだけではダメ

・例題からの応用力が問われる。

2 プログラミング教育の現状と難しさ



現状の高校でのプログラミング教育の実践

Python/JavaScript
等で分岐や繰り返し
を学習

アンプラグドで
並び替えなどの
アルゴリズムを学習

データサイエンスや
Webアプリの
実用プログラム開発

ロボットや
フィジカルな装置を
制御・計測

現状の高校でのプログラミング教育の実践

Python/JavaScript
等で分岐や繰り返し
を学習

アンプラグドで
並び替えなどの
アルゴリズムを学習

根本的に、検索や並び替え等のアルゴリズムをプログラミング
しているという実践はあまりないみたい

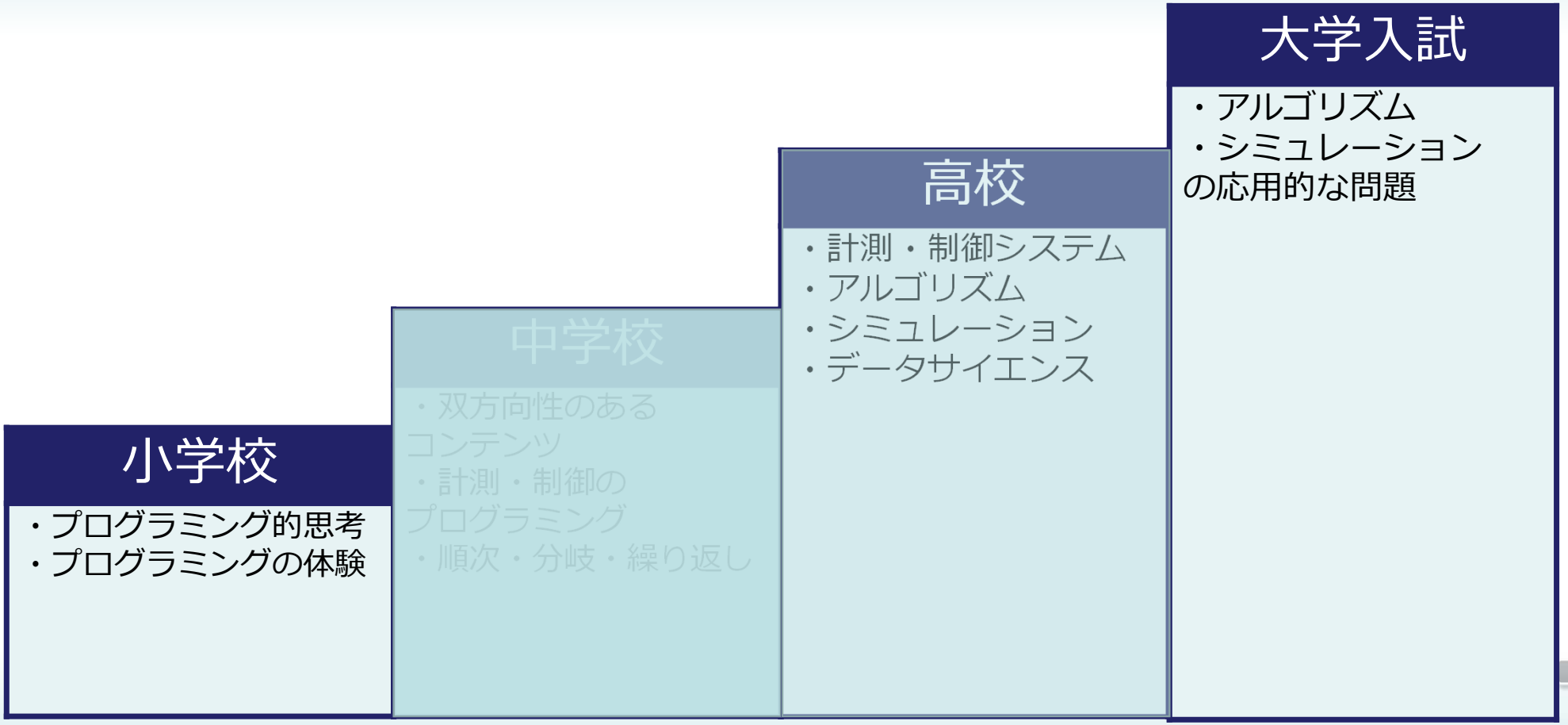
データサイエンスや
Webアプリの
実用プログラム開発

ロボットや
フィジカルな装置を
制御・計測

応用的な大学入試問題に対応できるような授業ではない。

高校プログラミングの問題: 小中高の連携の欠如

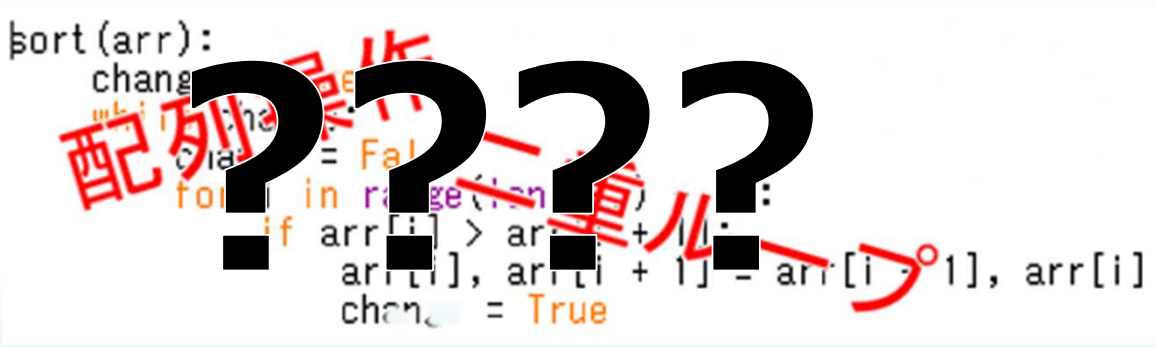
高校プログラミングの問題: 小中高の連携の欠如



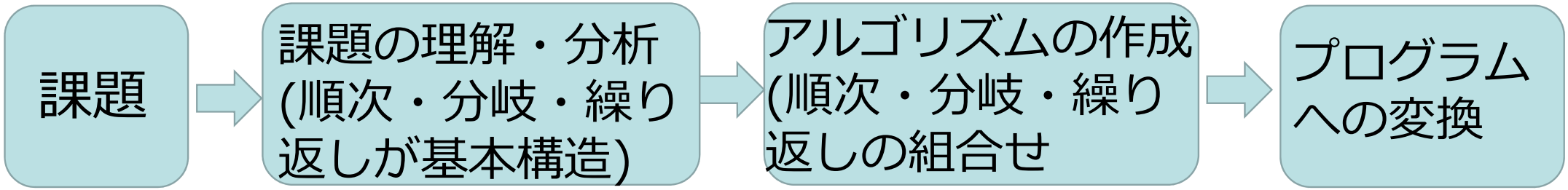
プログラミング自体の問題: 論理的な難しさ

情報関連の学部・学科においてアルゴリズムの
プログラミング教育は長年行われてきました。

その経験から配列・添え字による配列の操作・二重ループ等
が学生にとって難しいものと認識されています。



プログラミング教育のスタイル/プログラムの作成過程



フローチャート
UML

学習支援

問題:
問題はできても、他の新しい問題をプログラミングできない。

・デバッガやトレーサ等のプログラム作成支援ツール
・アルゴリズムアニメーション，変数の可視化ツール
・プログラムの自動判定のフィードバック
・穴埋め問題提示などのプログラムの作成補助ツール

3 プログラミングの認知過程を考える

3 プログラミングの認知過程を考える

1 2018年度の実践

- 変数の可視化やトレースなど考慮
- 個人ペースの協働学習
- アルゴリズムを中心に
- Scratchを使用

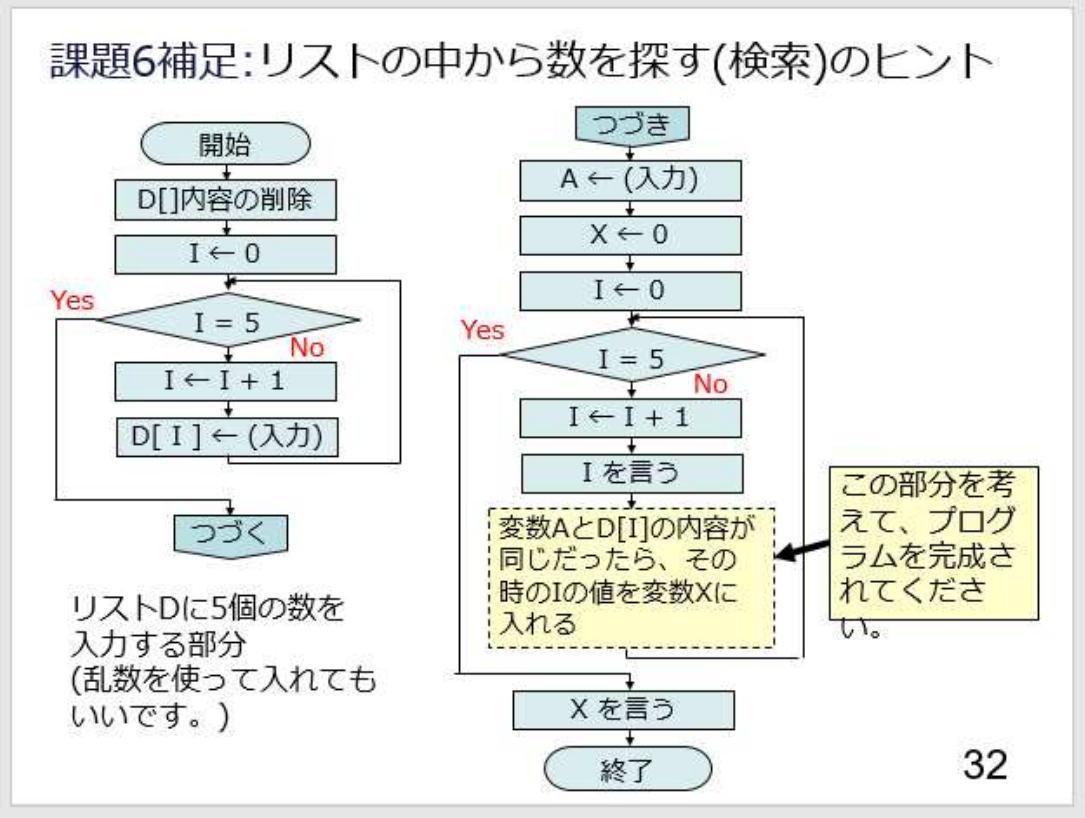
2 2019年度の実践

- プログラミングの認知過程に注目
- タンジブルな変数の利用
- 適切なステップ
- フローチャートとプログラムの対応

3 2020年度前半の実践

- 簡易的なタンジブル変数
- 初歩内容の充実
- より細かな指示の追加
- 動画解説の追加

フローチャートの限界

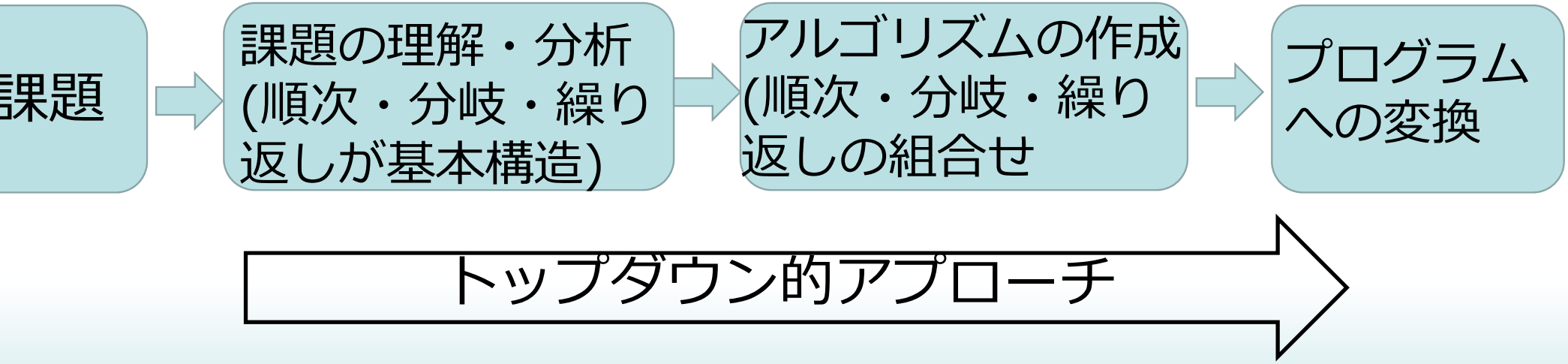


並び替えなどの二重ループをフローチャートで表現すると難しい。

フローチャートとプログラミングの対応が理解できない。

もっと認知過程に注目

「プログラミング学習において，何らかの認能力を伸ばそうとするのなら，そもそもプログラミングがどういう認知過程なのか，そこから調べていかなくてはならない」
(三宅,1987)



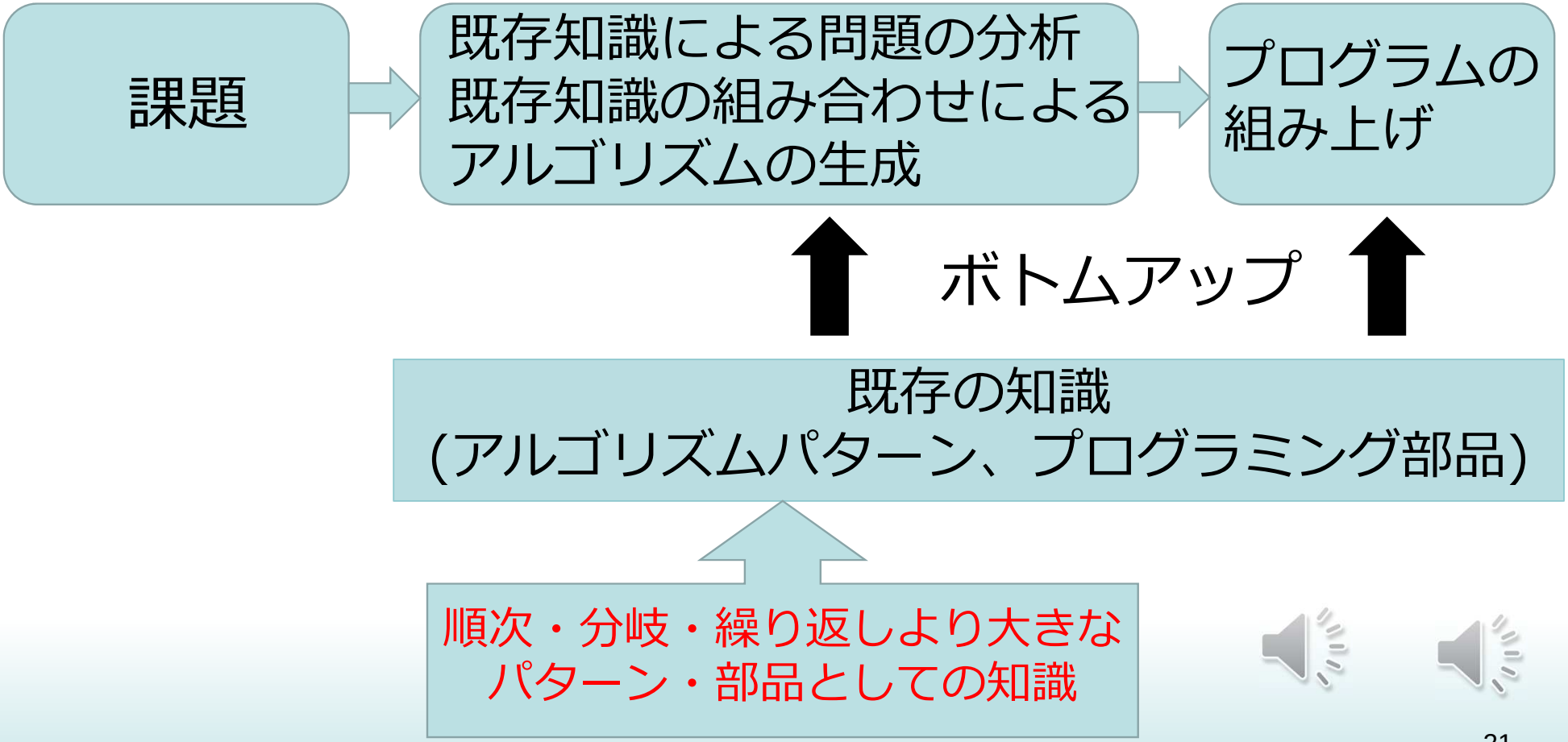
プログラミングの認知過程のヒント

子供のプログラミングを観察した中では，子供はアルゴリズムを考えて，それを実現するために個々の命令を使うよりは，ブロック(命令)を組み合わせた機能単位（プログラム部品）を利用することによってプログラミングを学習している。(太田ら, 2018)

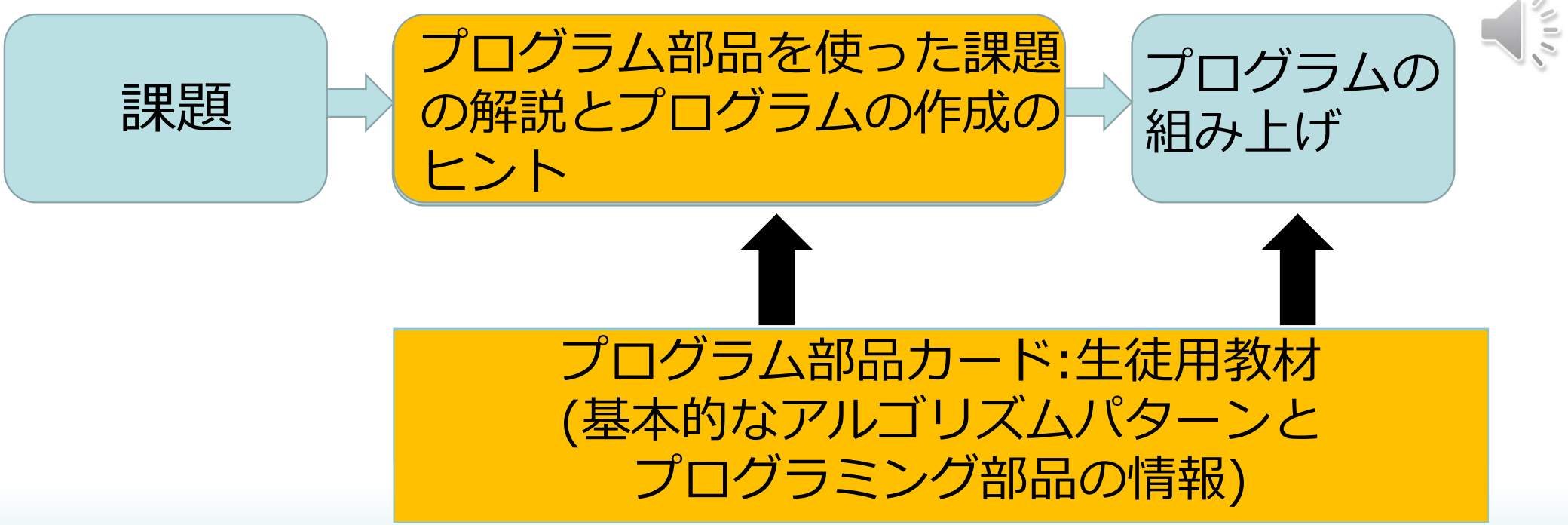
「競技プログラミングというのは(中略)，彼らは，問題を見た時に自分の頭に入っているアルゴリズムを応用できるかどか<<宝さがし>>のような感じだというのだ」(遠藤, 2021)

学習者が複数の命令をまとまった部品として認識することにより，プログラムの意味理解や作成に役立つことを指摘している(吉池ら, 2019)

既存知識によるボトムアップ的な問題解決



4 2020年度後半の教材と授業方法



プログラム機能部品

	プログラム機能部品
01	四則演算と変数への代入
02	キーボードから数値や文字の入力
03	条件を判断して，命令を実行
04	条件を判断して，〇と×で違う命令を実行
05	複数の条件を判断して違う命令を実行
06	数をカウントアップしながら繰り返す
07	リストの処理(取り出し，入れ替え，追加)
08	二つの変数の内容を入れ替える
09	繰り返しとリスト利用の組み合わせ
10	二重の繰り返し
11	二重の繰り返し(内の繰り返しの開始を変更)

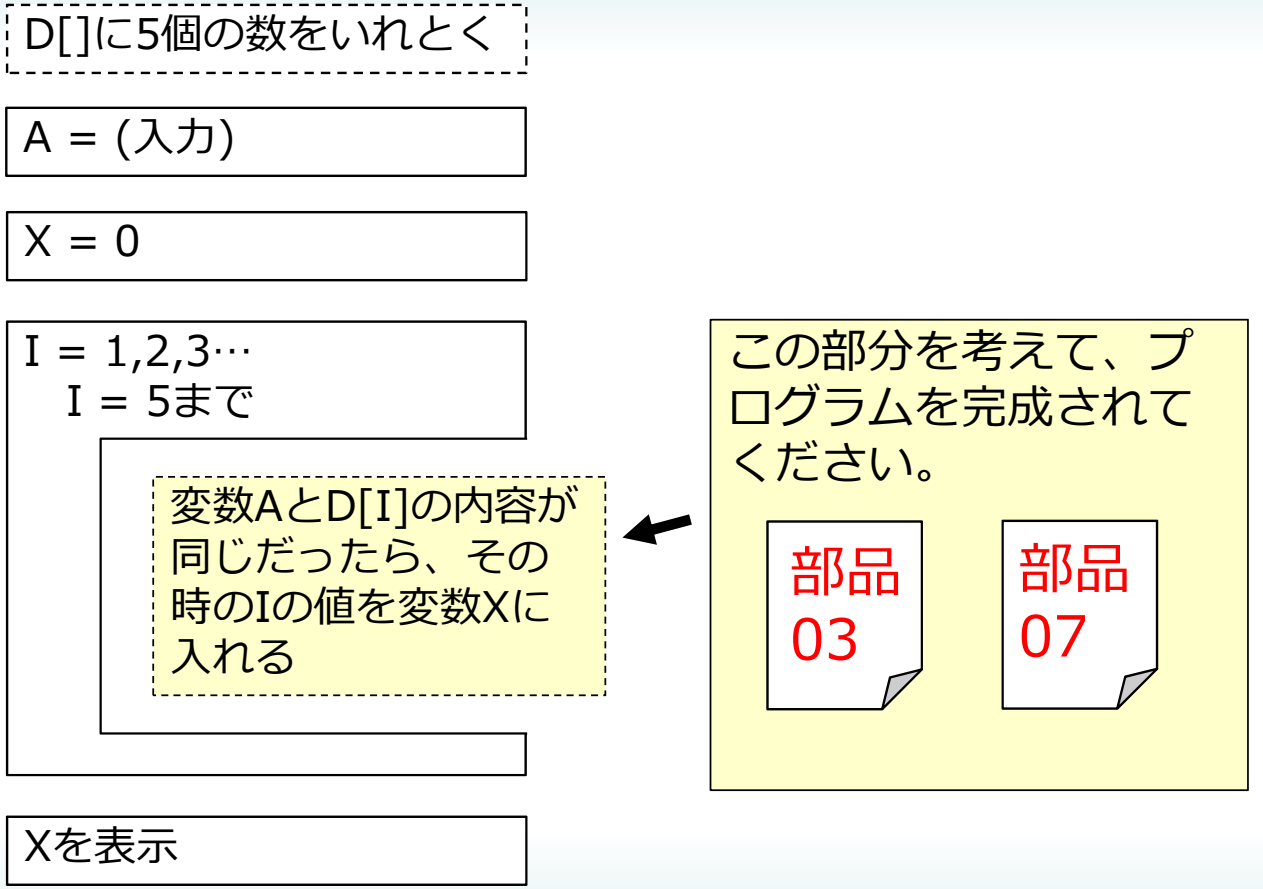
並べ替えのプログラムを作成するまでプログラム部品の定義

教材としてのプログラム機能部品カード

裏表に印刷したカード型教材として、個々の生徒に配布

生徒は指示書を見ながら課題のプログラミングを作成していく。

課題6補足:リストの中から数を探す(検索)のヒント



授業の形式:実践2020年度後半

	内容	チェック 理解	打込み	開発
1	変数とScratchでの利用			
2	Scratchでの変数への入力			
3	変数(X=X+1)			
4	プログラムの構造/フローチャート			
5	フローチャートとScratchの対応			
6	合格の判断			
7	一番簡単な自動販売機			
8	チャレンジ: 正三角形の判断			
9	1から10を言う			
10	単純な1から10の合計			
11	チャレンジ:単純な2からAまでの偶数の合計			
12	5回数字を入力してその合計を求めます.			
13	配列(リスト)を作る			
14	おくみくじを作る			
15	チャレンジ: 5回数字を入力してその合計(配列利用)			
16	配列に数をセットする方法			
17	チャレンジ: 配列の中から数を探す			
18	数の並び替えの作業の説明			
19	配列の中の一番小さい数を見つける			
20	配列の中の一番小さい数を配列の先頭に入れ替える			
21	二重繰り返しに挑戦			
22	最期のチャレンジ:数の並び替え			
23	発展課題1: 沢山の数の並び替え			
24	発展課題3: 並び替えの無駄を省く			43
25	発展課題2: バブルソート			

主要教材

・pdfの指示書(Web/スマホで見る)
・課題を示したチェックリスト

補助教材

・プログラム部品カード
・変数シート

チェックシートに使って個人ペースの学習

	No	内容	スライド No	チェック			
				[動画]	[理解]	[打ち込み]	[開発]
ここまでは、やってみよう。	1	学習の進め方	2		□		
	2	準備運動 1: Scratch の四則演算	4		□	□	
	3	準備運動 2: ネコに自分の名前を言わせる	5		□	□	
	4	プログラムと変数	6	□			
	5	課題 1: 入力した 2 個の数で四則演算	7				□
	6	打ち込み 1: 合格判断	8		□	□	
	7	課題 2: 合格不合格判断	9				□
	8	課題 3: 3 つの数の合計	10				□
	9	課題 4: 正三角形の判断	11				□
	10	打ち込み 2: 1 から 10 までの数を言う	12		□	□	
	11	プログラムに名前をつけて保存	13		□		
	12	打ち込み 3: 1 から 10 までの合計を言う	14		□	□	
	13	プログラムとフローチャート	15	□			
	14	変数/リストシートを使って考えよう	16		□		
	15	課題 5: 2 から X までの偶数の合計	17				□
	16	たくさんの数の合計の説明	19		□		
	17	打ち込み 4: リストを使った 3 つの数の合計	20		□	□	
	18	打ち込み 5: リストを使った 5 つの数の合計	22		□	□	
	19	リストと繰り返し	24	□			
	20	最後目標の課題の説明:数の並び替え	25		□		
	21	(休憩: おみくじ)	26		□	□	
	22	リストに数をセットする方法	28		□		
	23	課題 6: リストの中から数を探す	30				□
ここまで、できたらすごい。	24	課題 7: リストの中の一番小さい数を見つける	33				□
	25	課題 8: 一番小さい数を配列の先頭に入れ替える	35				□
	26	打ち込み 6: 二重繰り返しに挑戦	39		□	□	
	27	課題 9: 数の並び替え	43				□
もっと、やろう	28	発展課題 1: 他の並び替えの方法	46				□
	29	発展課題 2: 並び替えの方法の処理速度の違い	47				□

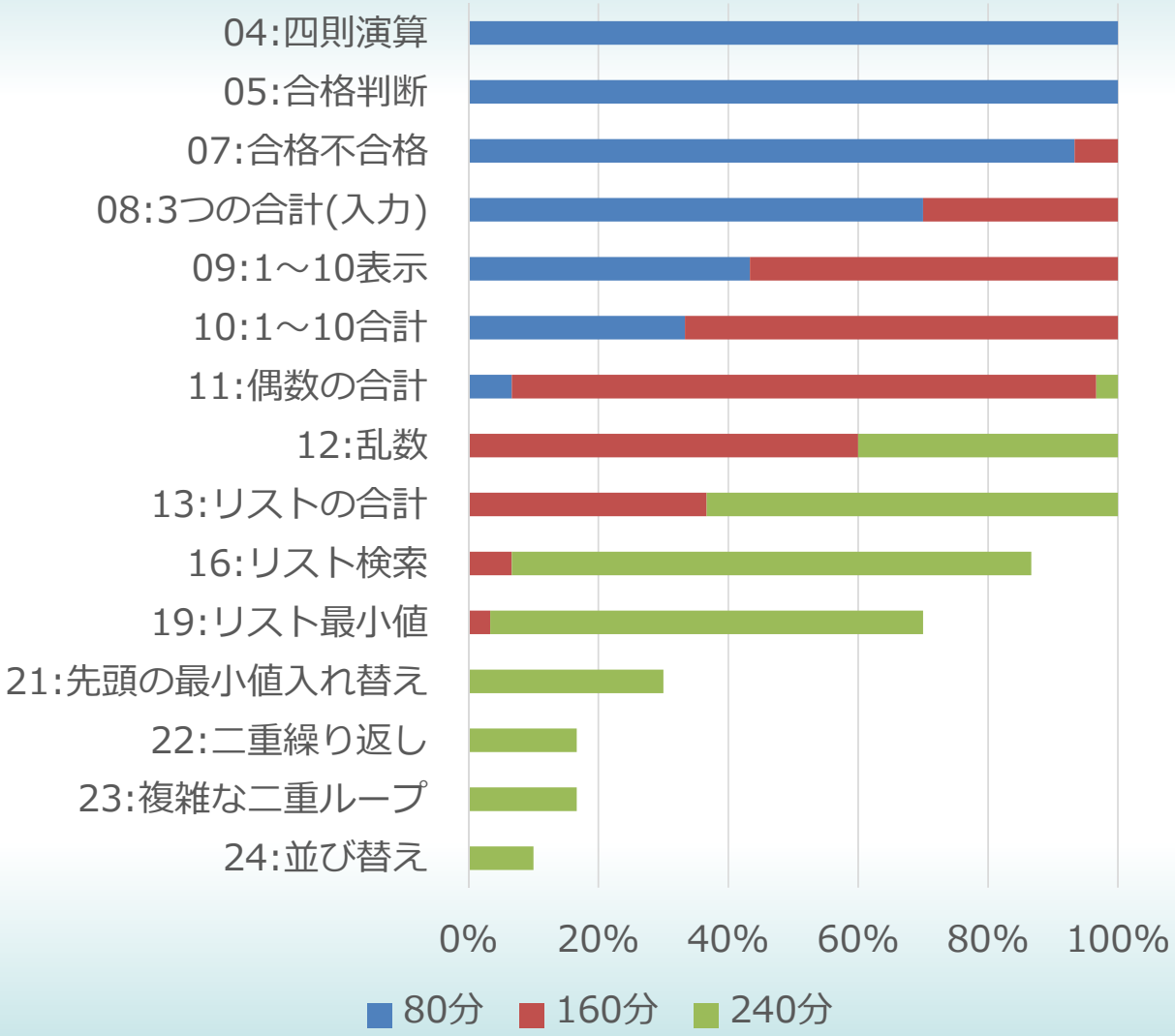
ファシリテータとしての教師の役割

従来の
一斉教授型

・支援の必要な生徒に対応する
・教えすぎないぐらいが丁度いいかも。

プグラミングの経験が少ない先生は、1年目は生徒のいっしょに勉強するぐらいの気持ちがいいかもしれません。

2020年度後半の実践結果



コロナ渦の短縮授業で
40分 x 6 回 = 240分

約10%の生徒が並び替えのプログラミングまで終了

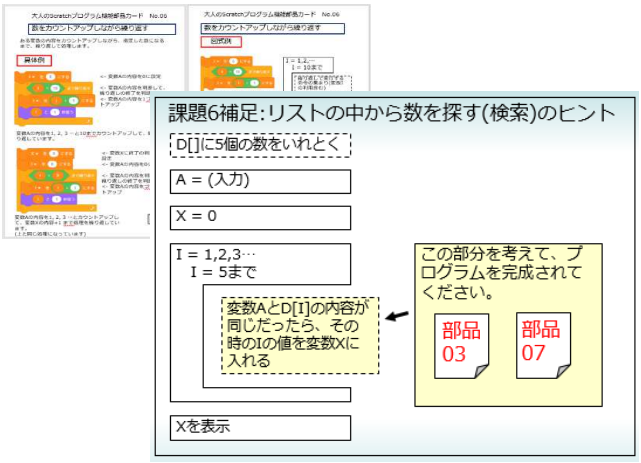
学習時間と他のScratchを使用した単元との関係

ゆるキャラ作り
(2時限)
・デジタルデータ
・デジタルデータの編集

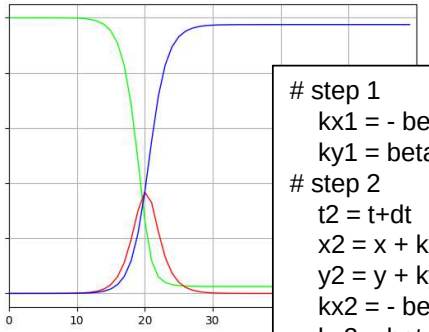
面白プログラミング
(2～3時限)
・情報デザイン
・プログラミング入門

おしえないアルゴリズムプログラミング教育(6時限)

5 大学入試に向けた今後の計画



情報I
アルゴリズムとプログラム
(Scratch)
2020年度に実践
2021年度用に改訂中




```
# step 1
    kx1 = - beta*x*y
    ky1 = beta*x*y - gamma*y
# step 2
    t2 = t+dt
    x2 = x + kx1*dt
    y2 = y + ky1*dt
    kx2 = - beta*x2*y2
    ky2 = beta*x2*y2 - gamma*y2
# update
    x = x + (kx1+kx2)*dt/2
    y = y + (ky1+ky2)*dt/2
    z = N - x - y
    t = t + dt
    cnt = cnt + 1
```

情報II
大学入試用
(Python)
大学入試対応の応用問題集
(開発中)

大学入試対応の応用問題集

基礎的な問題
基本
アプリケーション機能
平面での物体の操作
パズル問題
数学問題

応用的な問題
情報数学問題
ロボット制御
最適化/リソース割当
物理現象
グラフ/最短経路
予想/平均変化率
待ち行列
ライフゲーム
確率関係

大学の入試問題の分析に対応

例:ファーストフード店で、窓口でハンバーグを注文するまでの待ち時間をシミュレートするプログラムである。店員は一人で、注文が入ってから客に商品を渡すまで4分間かかるものとする。また、客は1～10分間隔でランダムに来店するものとする。このプログラムでは100人の客が来店した場合の平均待ち時間を計算する。

```
import random
arrival_time = 0
start_time = 0
end_time = start_time + 4
sum_wait = 0
for i in range(1, 100):
    arrival_time  += random.random() * 10
    start_time = max(arrival_time, end_time)
    end_time = start_time + 4
    sum_wait = sum_wait + (start_time - arrival_time)
    print (i, start_time, arrival_time)

print(sum_wait/100)
```

大学入試対応の応用問題集

基礎的な問題
基本
アプリケーション機能
平面での物体の操作
パズル問題
数学問題
応用的な問題
情報数学問題
ロボット制御
最適化/リソース割当
物理現象
グラフ/最短経路
予想/平均変化率
待ち行列
ライフゲーム
確率関係

例題の作成
Pythonでの
解答の作成
と検証

センター試験用
手順記述標準言
(DNCL)へ変換

例題の追加によ
る応用問題集の
作成とWeb公
開(2022年夏)

補足情報

2020年度に使用した教材等は下記のサイトからダウンロードできます。
なお、2021年度用の教材は2021年8月中に作成予定です。

高校「情報科」の教材・指導案作ってみました。
http://beyondbb.jp/

高校 情報科 教材	検索

Unit05 アルゴリズムとプログラム

Go.Ota