

高等学校「情報 I」における整列アルゴリズムの可視化プログラミング

稲川 孝司

喜家村 奨

西野 和典

高橋参吉

帝塚山学院大学

帝塚山学院大学

太成学院大学

NPO 法人学習開発研究所

t-inagawa@tezuka-gu.ac.jp

susumu@tezuka-gu.ac.jp

k-nishino@tgu.ac.jp

takahasi-san@u-manabi.org

小・中・高のプログラミング教育の継続性の問題点は、プログラミング言語の接続性であるといえる。そこで、ブロック型とテキスト型のどちらでもプログラムを作成できる micro:bit に 128×64 ドットのグラフィックが表示できる OLED の拡張ディスプレイを接続し、情報 I のアルゴリズムとプログラミングの単元で整列アルゴリズムを理解するための可視化プログラムを作成した。

1. はじめに

学習指導要領が改訂され¹⁾、小学校では令和 2 年から中学校では令和 3 年から全面実施され、高等学校では令和 4 年度から学年進行で実施されている。授業内容については、Society5.0 に向けた人材育成の推進ということで、小・中・高を通じて、情報活用能力を言語能力と同様に「学習の基盤となる資質・能力」と位置付けて育成することをめざしている。またプログラミング教育については、発達段階に応じて充実を図るために、小学校から必修化され、中学校では「技術・家庭科」においてプログラミングに関する内容が充実され、高等学校では必修科目「情報 I」に「プログラミング」の単元が新たに必修となっている。

高等学校学習指導要領解説情報編²⁾によると、共通教科情報科の学習内容は、中学校技術・家庭科技術分野の内容「D 情報の技術」との系統性を重視し、その指導を行うためには、中学校技術・家庭科技術分野の改善内容を十分踏まえることが重要であることが示されており、教員研修教材にボード型コンピュータの教材を使った事例が示されている。

そこで、高等学校情報 I のアルゴリズムとプログラミングの単元で、micro:bit に拡張ディスプレイを接続して整列のプログラムを作成し、アルゴリズムの可視化の教材を作成した。

2. アルゴリズムとプログラミング

2.1 アルゴリズムとデバッグ

プログラミングの授業で大切なことは、問題解決のためのアルゴリズムを考え、変数や配列、リスト、関数の概念を学び、プログラムを作成して動作させ、その実行結果からプログラムの動作を理解することである。その中でも基本的なものが「ソート（整列、並び替え）」である。

情報 I の教科書では、アルゴリズムの例として「整列」を取り上げており、交換法や、選択法など

の処理の方法を具体的に説明し、それらを実現するためのプログラムも記載されている。

しかし、プログラミングの初心者には、ソースコードからプログラムの流れや実行時の変数の値の変化がわからないため、プログラムの動作を正確に把握することは、困難である。

一般的には、デバッガを用いればプログラム実行時の変数の値を調べることが可能であるが、デバッガの操作やブレイクポイントの設定が必要になるので、初心者には難しい。そこで、micro:bit に拡張ディスプレイを接続し、プログラムの動作の可視化を試みた。

2.2 micro:bit と拡張ディスプレイの利用

micro:bit では 5×5 の LED を用いて様々な結果を表示させることができるが、グラフを表示するといったことでは大きな制約を受ける。そこで、本研究では、拡張ディスプレイ（Kitronik View Graphics OLED 128×64）（図 1）を使用し、グラフ化したデータで動作を可視化している。この拡張ディスプレイは 128×64 ドットの解像度を持っており、単色ではあるが文字を表示させたり、点や線を描かせたりできる。



図 1 拡張ディスプレイによる動作の可視化



図2 拡張ディスプレイ上に配列内の64個の数値を棒グラフ形式で表示させるプログラム

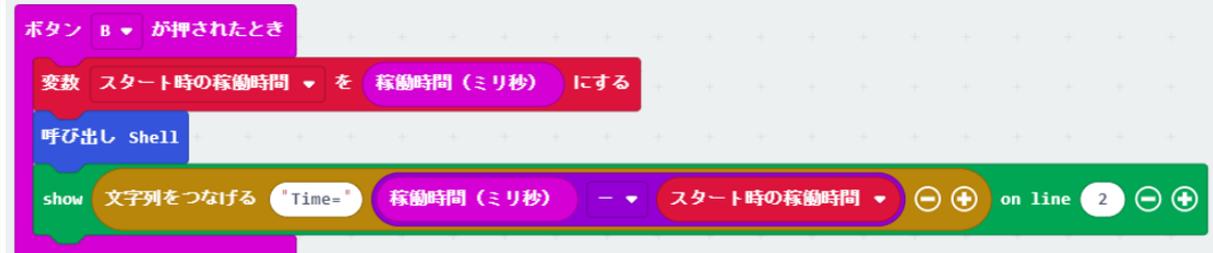


図3 内部タイマーを使ったソートの稼働時間の計算方法とその表示プログラム

3. 整列プログラムの動作の可視化

整列プログラムの動作を視覚化する方法として、配列の要素の値をそのまま表示してソートの過程を見せる方法がある。しかし、micro:bit単体での表示が貧弱なため、拡張ディスプレイのグラフィック機能を利用した可視化プログラムを考えた。

その方法は、配列を用意して、ソートするデータを最大値50の数までの乱数として64個作成し、1ドットずつ離して数値を棒グラフの長さとして、拡張ディスプレイの二次元座標上で線を描くものである(図2)(実際の動作画面は図1を参照)。

また、内部タイマーを使ってソートの処理時間を計算し、ソート終了時に処理時間を表示するプログラムを追加している(図3)。

図4は交換法(バブルソート)のプログラム例である。配列内の64個のランダムな数字に対し、j番目とj+1番目を比較し、j番目がj+1より大きければ2つの数字を入れ替えるようにしている。それをj=0から63まで順に繰り返して、並び替えをおこなっている。



図4 交換法のプログラム例(一部)

図5は挿入法のプログラム例である。1番目のデータに対し、それ以降のすべてのデータと見比

べて、小さければ置き換える。次に2番目のデータに対しそれ以降のすべてのデータと見比べて、小さければ置き換える。これを繰り返して、並び替えをおこなっている。

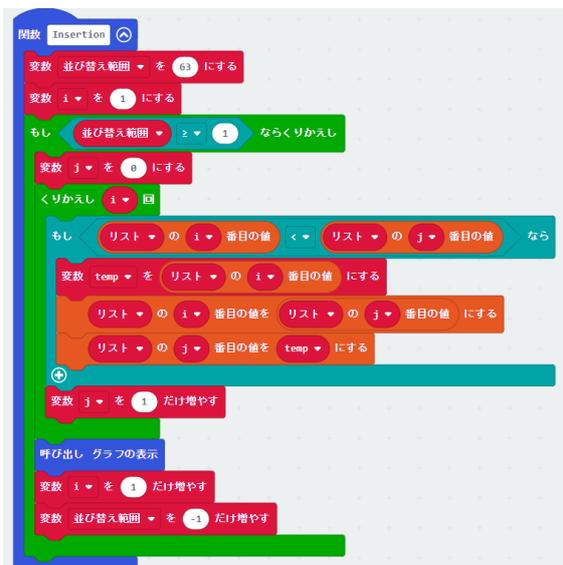


図5 挿入法のプログラム例(一部)

4. おわりに

micro:bitと拡張ディスプレイを使って、アルゴリズムとプログラミングの単元で、整列のアルゴリズムを理解し、それを可視化するプログラム教材を作成した。ブロック型とテキスト型を併用してプログラムを作成し、拡張ディスプレイを使うことで整列のアルゴリズムの動作の可視化が可能になり、単元の理解が深まる。

本研究は2020年度JSPS科研費JP20K02528の助成を受け、またNPO法人学習開発研究所の三輪吉和氏から助言をいただき、感謝いたします。

参考文献

- (1) 文部科学省:平成30年度公示高等学校学習指導要領, 東山書房(2019).
- (2) 文部科学省:高等学校学習指導要領(平成30年公示)解説 情報編 開隆堂出版(2019).