

## 修士論文の和文要旨

研究科・専攻	大学院 情報理工学研究科 情報ネットワーク・工学専攻 博士前期課程		
氏名	関下堅也	学籍番号	2031090
論文題目	探索および整列アルゴリズムの学習支援教材の提案と評価		
要旨	<p>日本では論理的思考力の育成を目的にプログラミング教育と情報の教育に着手しており、高等学校では情報 I を必修化する運びであるが中山らの調査によると必ずしも情報学についての幅広い知識や技術をもつ者が情報科の教科担任となるわけではなく、教員不足や地域格差も確認されている。</p> <p>情報 I の高等学校教員研修用教材によると、探索、整列アルゴリズムが取り扱い範囲にあり、アルゴリズムの動きだけではなく、実際に動かしたときに計算量が最悪となるパターンも取り扱うこととしているが、教員が習得および学生へ教授する難易度は容易くないと思われる。そのためか教科書によっては探索、整列アルゴリズムについて触れない教科書があり、高校生に学校の選定した教科書による学習到達度の差が発生する恐れがある。</p> <p>情報 I は 2025 年の大学入学共通テストに導入予定で、想定問題によるとアルゴリズムの擬似言語の問題が出題予定であり、過去のセンター試験問題を見ると 2008 年の情報関係基礎においてバブルソートの問題が出題されたこともあり、受験においても対策の必要がある。現在、教科書ではどのようにアルゴリズムが動くかをアニメーションで見る補助教材があるのみで、アルゴリズムの学習に対して支援が必要と考えられる中、情報に関する学習を支えるシステムを取り扱う情報処理学会の CE 研究会内ではアルゴリズムに関する学習教材は他の分野よりも支援が少ないことが課題となっている。</p> <p>本研究では先行研究より得られた結果からアルゴリズムの動作を詳しく説明した上で手作業によってアルゴリズムを体験した際の動作手順をログとして保存する、ログに対して共通動作をマークする、ログを抽象化することで擬似言語化するシステムを構成することで、学習者のアルゴリズム分析能力と教科書による学習到達度の差をなくす、教師の負担が増えないように本システムで学習が完結するという目的を達成できるか評価した。</p>		

令和3年度 修士論文

# 探索および整列アルゴリズムの学習支援教材の提案と評価

所属	情報・ネットワーク工学専攻
学籍番号	2031090
氏名	関下 堅也
指導教員	中山 泰一 教授
副指導教員	寺田 実 准教授
副指導教員	武石 典史 教授

# 目次

第1章	はじめに	1
1.1	本研究の背景	1
1.2	本研究の目的	2
1.3	本論文の構成	2
第2章	関連研究	3
2.1	アルゴリズムの学習過程についての先行研究	3
2.1.1	プログラミング初学者にむけたアルゴリズム的思考習得のための学習支援手法	3
2.1.2	アルゴリズム構造に基づくコア・イメージのアルゴリズム教育への応用	3
2.1.3	アルゴリズム構築能力育成の導入教育: 実作業による概念理解に基づくアルゴリズム構築体験とその効果	4
2.2	アルゴリズムの学習教材について	4
2.2.1	Teaching and learning computer science sorting algorithms with mobile devices: A case study	4
第3章	システムの実装	6
3.1	学習コンテンツとして取り上げたアルゴリズム	6
3.2	段階別の学習について	6
3.2.1	1段階目:アルゴリズムの全体像を掴む(手作業によるアルゴリズムの体験)	6
3.2.2	2段階目:アルゴリズムの共通動作を認識する(アルゴリズムの体験ログに分岐点を設ける)	8
3.2.3	3段階目:アルゴリズムのコンピュータ上での表現	8
3.3	確認テスト	13
3.4	実力テスト(挿入ソート)	13
第4章	実験	15
4.1	実験スケジュールと被験者数	15
4.2	実験の目的と調査内容	15
4.3	評価方法	15
4.3.1	評価の流れ	15
4.3.2	制御構造の説明について	16
4.3.3	アルゴリズムの学習について	16

4.3.4	確認テスト	17
4.3.5	実力テスト(挿入ソート)について	17
4.3.6	総合評価	18
<b>第5章</b>	<b>実験結果</b>	<b>19</b>
5.1	制御構造の説明について	19
5.2	バブルソートの学習	19
5.2.1	1段階目	19
5.2.2	2段階目	19
5.2.3	3段階目	19
5.2.4	アンケートの結果	20
5.3	二分探索の学習	20
5.3.1	1段階目	20
5.3.2	2段階目	21
5.3.3	3段階目	21
5.3.4	アンケートの結果	21
5.4	確認テスト	21
5.4.1	問1:バブルソートの状態を示す問題	21
5.4.2	問2:二分探索の実行回数	21
5.4.3	問3:二分探索と線形探索の比較の選択問題	23
5.4.4	アンケートの結果	23
5.5	実力テスト(挿入ソート)	23
5.5.1	問1:挿入ソートの試行回数を問う問題	23
5.5.2	問2:挿入ソートの段階ごとの状態を示す問題	23
5.5.3	問3,問4:挿入ソートの自然言語及び疑似言語の選択問題	24
5.5.4	アンケートの結果	24
5.6	システム総評のアンケート	24
<b>第6章</b>	<b>考察</b>	<b>26</b>
6.1	段階別学習の1段階目について	26
6.2	段階別学習の2段階目について	26
6.3	段階別学習の3段階目、実力テストの問3、問4について	27
6.3.1	誤答について	27
6.3.2	解答時間について	28
6.3.3	アンケートについて	29
6.4	確認テストについて	29
6.5	実力テストについて	29
6.6	総評のアンケートについて	29
<b>第7章</b>	<b>終わりに</b>	<b>30</b>
7.1	まとめ	30
7.2	今後の課題	30

付 録 A 1 段階目のバブルソートの説明と用いた図	34
付 録 B 1 段階目の二分探索の説明と用いた図	36
付 録 C 実力テストの挿入ソートの説明と用いた図	38

# 第1章 はじめに

## 1.1 本研究の背景

日本では論理的思考力の育成を目的にプログラミング教育と情報の教育に着手しており、高等学校では情報Ⅰを必修化する運びであるが中山らの調査によると必ずしも情報学についての幅広い知識や技術をもつ者が情報科の教科担任となるわけではなく、教員不足や地域格差も確認されている [1]。

情報Ⅰの高等学校教員研修用教材によると、探索、整列アルゴリズムが取り扱い範囲にあり、アルゴリズムの動きだけではなく、実際に動かしたときに計算量が最悪となるパターンも取り扱うこととしているが [2]、教員が習得および学生へ教授する難易度は容易くないと思われる。そのためか教科書によっては探索、整列アルゴリズムについて触れない教科書があり、高校生に学校の選定した教科書による学習到達度の差が発生する恐れがある。

情報Ⅰは2025年の大学入学共通テストに導入予定で [3]、過去のセンター試験問題を見ると、2008年の情報関係基礎においてバブルソートの問題が出題されたこともあり [4]、受験においても対策の必要がある可能性がある。現在、教科書ではアルゴリズム図鑑 [5] でどのようにアルゴリズムが動くかをアニメーションで見る指導形式である。長瀧ら [6] の調査によると、情報に関する学習を支えるシステムを取り扱う情報処理学会のCE研究会内ではアルゴリズムに関する学習教材は他の分野よりも支援が少ないことが課題となっている。

アルゴリズムに関する教材が不足している一方でプログラミングに関する研究は進んでおり、例えばオブジェクト指向言語の学習支援であれば恒川ら [7] のメンバの可視化、相馬らの [8] クラス図の表示、小清水ら [9] のクラス図の作成補助などが挙げられる。これらの研究は可視化による学習支援を行っており、アルゴリズムの学習においてもどのような処理が起こっているかを可視化することが支援につながるのではないかと考えられる。また、鈴木ら [10] の研究と安達ら [11] の研究によると学習の支援に当たり、プログラミングの学習には問題文とソースコードの記述に移る間の処理手順のまとめである構成要素を考えるプロセスがあり、このアルゴリズムの構成でつまづくとの声もある。とくに鈴木らは構成要素も問題文に近い構成要素、ソースコードに近い構成要素があるとしている。同様に川口 [12] は、図や式などの中で特にアルゴリズムの構造に関わるものを「コア・イメージ」と呼んでいる。杉浦ら [13] は、アルゴリズム構築能力の育成には教具を用いて手作業でアルゴリズムを体験し、体験をそのままプログラミングできるシステムを開発することが重要であると、学習支援効果を得た。これを受けて浅野ら [14] は発見学習として配列に関するアルゴリズムの部品となる基本操作を学習させて基本操作の組み合わせで既存の整列アルゴリズムを発見させる手法を取ることで学習成果を得た。間辺ら [15] は同

様の手法を取った上で発見したアルゴリズムを発表することで思考の整理をする効果について言及していた。このことから手作業でアルゴリズムを経験し、基本的な動作を理解することと思考の整理を行うことが大切であると言える。

これまでの学習手法はアルゴリズムの理解に必要な要素を学んでからアルゴリズムを説明してプログラミングへ移行する手法が一般的であるため他の手法での教育を調査したところ、大田 [16] は先に既存のアルゴリズムの理解と分析をし、既存知識でプログラムを組み上げる学習手法を提案し、学習効果が得られた。

そこで本研究では著者 [17] の発表から発展し、3段階の学習構造によって探索、整列アルゴリズムを学習する教材を開発し、実験を行った。1段階目はアルゴリズムの中で重要な処理手順を文章と図を用いて解説して動作のイメージを明確にもたせた上で手作業によってアルゴリズムを経験してもらい、どのような作業をしたかのログを取る。2段階目はログに対して処理が切り替わる部分に対してマーカーをつけることでアルゴリズムが規則的に繰り返している処理に注目してもらい、3段階目はマーカーをつけたログをもとに、アルゴリズムの抽象化に必要な単語を定義して図で説明し、繰り返した回数や整列アルゴリズムにおいて交換が発生する条件などをログから分析し、定義した単語から抽象化をすることを選択問題形式で自然言語によって行う。この自然言語によって抽象化した文章を参考にしながらセンター試験用手順記述標準言語 (DNCL)[18] のような疑似言語の選択問題を解いてもらうことでコンピュータ上でのアルゴリズムの表現を学んでもらうという構成によって学習効果を測る。

## 1.2 本研究の目的

本研究の目的は以下のとおりである。

- 新学習指導要領の施行に伴って小学校、中学校でビジュアル型プログラミングを学んだ初学者にあたる生徒が高校で情報Ⅰを履修した際に探索、整列アルゴリズムの処理手順を理解することを通してアルゴリズムの分析能力を身につけること。
- 上記によってアルゴリズムの取り扱いの有無に関わらず教科書による学習到達度の差をなくすこと。
- 教師の負担とならないよう、本システムで学習が完結できるように設計すること。

## 1.3 本論文の構成

本論文は、本章で背景や目的の説明、2章で関連研究の紹介、3章でUIの実装、4章で実験、5章で実験結果、6章で考察、7章でまとめと今後の課題の構成となっている。

## 第2章 関連研究

### 2.1 アルゴリズムの学習過程についての先行研究

#### 2.1.1 プログラミング初学者にむけたアルゴリズム的思考習得のための学習支援手法

鈴木ら [10] は、プログラミング初学者が言語の特徴やアルゴリズムの動作を理解していても、実際にプログラムのソースコードとして記述する前に、問題文とソースコードの間にあるどのような処理をどの順番で行うかを考える「構成要素」の考え方を提唱した。構成要素を理解している人は、問題文の言い回しから必要な要素を発見し、それを満たす機能の作成とデバッグを繰り返すことでプログラムに不足がないかを確認していると考えた。

この構成要素も「問題文に近い」短い文章の組み合わせによって処理手順をまとめたもの、「ソースコードに近い」抽象的な処理手順でプログラム構成を表すものに分かれており、大枠でプログラムの流れを捉えて、具体的な手順を考えることで初学者が問題文からプログラムを作成できると仮定し、システムを開発して実験を行った。

システムはアルゴリズム内の処理を短い文章で分割したブロックを用意して並び替えて処理の流れ図を作成する機能、並び替えた各ブロックの短い文章に対し、具体的に変数や代入部分を選択して埋めていく穴埋め問題の機能を用意し、問題を3問用意した。

このシステムを使用して問題を解いてもらってアンケートを取った結果、処理の流れを考えることで問題が解きやすくなったと解答したため一定の学習効果が得られた。

本研究ではこの構成要素を並び替えなどで問題化はせず、構成要素を図で明示化した上で手作業でアルゴリズムを体験してもらう手法を取った。

#### 2.1.2 アルゴリズム構造に基づくコア・イメージのアルゴリズム教育への応用

川口 [12] はアルゴリズムを学ぶ意義は問題解決の手順としての理解だけではなく、ベースとなるものの捉え方や考え方を含めたアルゴリズムの構造を理解することが重要であると考えた。そこでアルゴリズム名を聞いたときに想起される図、式、グラフ、文章とその組み合わせの中で特にアルゴリズムの構造に関わるものを「コア・イメージ」と呼び、これを理解させる授業を大学で試みた。

授業において、川口は探索、整列、再帰のアルゴリズムに関係するものだけでなく、 $n$ 進法や濃度計算なども構造を本質的に理解できるように文章や表を図に落とし込んで説明を行った。



その結果、コア・イメージを用いることでアルゴリズムや再帰構造などが視覚的に理解できた、数学などの他の教科で苦手意識を持った学生が問題を解けるようになったなどの本質的な構造を理解することで学習効果が得られた。

本研究で図を作成する際にコア・イメージの考え方が伝わるようにした。

### 2.1.3 アルゴリズム構築能力育成の導入教育：実作業による概念理解に基づくアルゴリズム構築体験とその効果

杉浦ら [13] は、与えられた課題を理解・分析してそれを手順に分解・詳細化してプログラムとして記述可能なアルゴリズムを組み立てるアルゴリズム「構築能力」をどのように育成するかが課題であるとした。

アルゴリズム概念を理解するにはプログラミング言語の読解ではなく手作業でアルゴリズムを理解すること、アルゴリズムを組み立てる能力を育成するためにはプログラミング言語を記述する学習コストを減らし、アルゴリズムの構築に注力できる環境が有効であると先行研究より判断し、連携を測る工夫を行った。

まず学習者に手作業でアルゴリズムを実行させて理解させ、その後に手作業の結果を忠実にプログラミングさせることのできる環境を構築した。手作業に数字が記載されたカードなど現実世界の教具を用いることによってプログラミング環境に対して教具をデータの型にする、アルゴリズム記述とプログラミング言語による処理の表現の乖離を少なくした。

その結果被験者は、普段の生活で気にしていなかった手順を意識すること、流れをはっきり整理することがアルゴリズムを理解することにつながると認識することができ、実際にアルゴリズムを構築にも必要であることを理解できた。また、整列・探索アルゴリズムの作成に関しても構築難易度の高いアルゴリズムをプログラミングできたため学習効果が得られた。

本研究では手作業による体験の重要性を認識し、パソコン上でアルゴリズムを体験して得られた手順のログを用いて学習に反映することとした。

## 2.2 アルゴリズムの学習教材について

### 2.2.1 Teaching and learning computer science sorting algorithms with mobile devices: A case study

Boticki ら [19] は android スマートフォン端末を利用するソートアルゴリズム学習コンテンツを開発した。普通の授業や学習において紙とペンを使うが、その場合要素をソートする手順が適切ではないことで本来学ぶべきアルゴリズムを学べない恐れがあると考えた。

そこでソートアルゴリズムを正しく学べる支援をするために場所を選ばずに実行できる web 形式で開発した。アニメーションを使う際は学習の妨げにならないように自分で設計したアルゴリズムと正解のアルゴリズムの2つのアニメーションを流せる機能、学生のソーティングを支援するためにループの継続条件やエラーの修正方法をテキストメッセー

ジで表示する機能、学生のプログラミングの正確さと速さとアルゴリズムの難易度に依存する得点方式を提供する「sortko」というアプリケーションを開発した。

アンケートの結果、ソートアルゴリズムは大学生にとって初めて触る内容であること、学生も PC やスマートフォンでアルゴリズムを学ぶことができていることがわかった。また、試験においてクイックソートの課題を解かせたところアプリケーションの使用頻度と 0.47 という相関が得られたため学習効果が得られた。

本研究では手作業でアルゴリズムを体験するとき間違えてアルゴリズムを学習しないようにする工夫と、アニメーションとは異なるが体験画面において正しい選択した際に値の色を変えるように参考にした。

## 第3章 システムの実装

実装したシステムは web ベースとなっており、それぞれの機能を説明をする。

### 3.1 学習コンテンツとして取り上げたアルゴリズム

今回は線形探索、二分探索、単純ソート、バブルソートを実装し、特に二分探索とバブルソートは配列の頭から順番に比較や交換をしていく線形探索、単純ソートに比べると学習する難易度が高いためより注力した。これらは3つの教科書会社の教科書において手順の説明やフローチャート化、コード化されていたアルゴリズムであるため本論文において学習コンテンツとして採用した。

### 3.2 段階別の学習について

アルゴリズムの全体像を掴むこと、全体像より共通の動作を認識することによってアルゴリズムの表現に必要な変数を定義してその変数でアルゴリズムの動作を自然言語も交えて表現する抽象化と、DNCL による擬似言語化をすることができるようになることを考えた。そのため3つの段階を用いることで学習者のアルゴリズムの理解をより深められるようにした。

本システムではアルゴリズムの動作の学習と、学習したアルゴリズムを理解できているかの確認テスト、習っていないアルゴリズムの説明を見て動作の理解と抽象化、擬似言語化ができるかを測る実力テストを用意した。

#### 3.2.1 1段階目:アルゴリズムの全体像を掴む(手作業によるアルゴリズムの体験)

実際に正しいアルゴリズムを体験するために、アルゴリズムの説明と別のウィンドウでアルゴリズムの動きの説明をした図を表示できるようにした上で、ボタン形式で配列の中身や大小関係、交換の有無などを押して操作をしたことを自然言語でログとして表示し、ログの部分を読み取ることで正しい選択肢に誘導できるようにした。web ページの構成は図 3.1 に示す。アルゴリズムの説明に用いた図は付録に示す。

バブルソートの学習において図 3.2 が操作前の UI を示しており、図 3.2 に「3 を押す、5 を押す」を行うと図 3.3 になる。ログを見ると次に大小を比較するボタンを押すように誘導をしている。

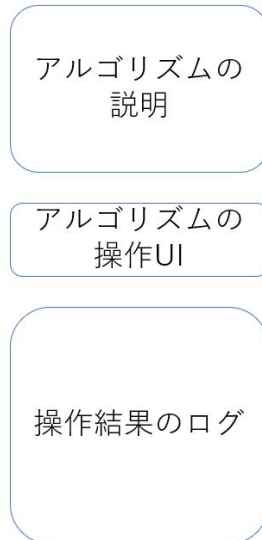


図 3.1: 1 段階目の web ページ構成

実際にバブルソートを行ってみよう。

3	5	2	1	4
よりも大きい		よりも小さい		

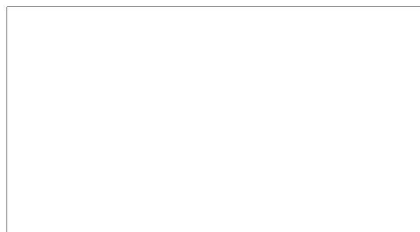


図 3.2: バブルソート操作前

実際にバブルソートを行ってみよう。

3	5	2	1	4
よりも大きい		よりも小さい		

左の値は3。  
右の値は5。  
左の値が右の値よりも

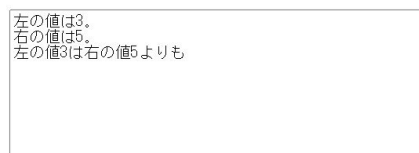


図 3.3: バブルソート操作後

同様に二分探索における操作前のUIを図3.4に、図3.4で「1を押す、18を押す、9を押す」を行った状態を図3.5に示した。ログによって次に大小関係に関わるボタン、それを押した後に配列の左右のどちらに注目するかを示すログが表示される。図3.3と図3.5にあるように、配列のどこを操作しているかをわかりやすくするために両アルゴリズムにおいて押した部分の色分けができるようにした。アルゴリズムの説明に用いた文章や図は

実際に「4」を二分探索で探してみよう。

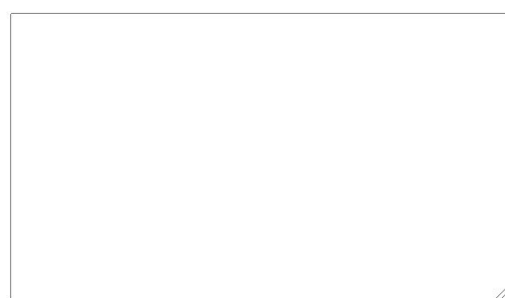
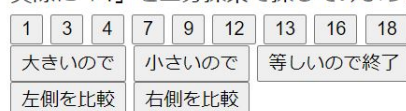


図 3.4: 二分探索操作前

実際に「4」を二分探索で探してみよう。

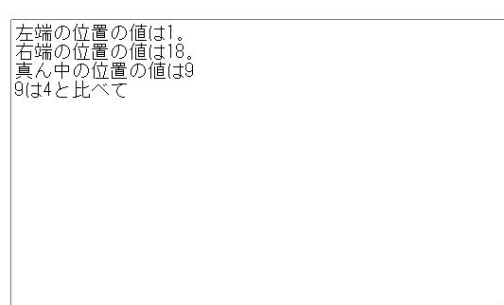
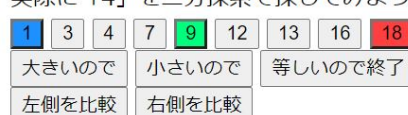


図 3.5: 二分探索操作後

付録にまとめる。

このシステムでは正誤の判定が表示され、押したボタンと正誤が保存される。

### 3.2.2 2段階目:アルゴリズムの共通動作を認識する(アルゴリズムの体験ログに分岐点を設ける)

1段階目で得られたログに対して1回のループ内での処理が終わった地点に●を挿入してもらうことで共通の操作を行っていることや二重ループであれば処理が減ることで●同士が近づくことに気づいてもらえるようなUIを設けた。このシステムでは正誤の判定が表示され、●を挿入した位置と正誤が保存される。

図3.6にバブルソートの場合と解答を示す。今回1段階目の体験で得られたログに対し、13行目、23行目、30行目、34行目に●を挿入することを正解とした。●同士の間隔は13行、10行、7行、4行と減ってきて近づいて来ているため、バブルソートにおいてはループごとに配列の要素の比較と交換の回数が減少していることを示唆できていると考えた。

図3.7に二分探索の場合と解答を示す。こちらでは●を5行目、9行目、13行目に挿入する。●同士の間隔は変わらないため共通の操作を行っていることを示唆できていると考えた。

### 3.2.3 3段階目:アルゴリズムのコンピュータ上での表現

まずはじめにアルゴリズムの表現に必要な変数を定義し、1段階目で得られたログから定義された変数による抽象化を選択形式で文章を穴埋めることを行うことができる。これ

01:	●	左の値は3。	●削除
02:	●	右の値は5。	●削除
03:	●	左の値3は右の値5よりも小さいので交換しない。	●削除
04:	●	左の値は5。	●削除
05:	●	右の値は2。	●削除
06:	●	左の値5は右の値2よりも大きいので交換する。	●削除
07:	●	左の値は5。	●削除
08:	●	右の値は1。	●削除
09:	●	左の値5は右の値1よりも大きいので交換する。	●削除
10:	●	左の値は5。	●削除
11:	●	右の値は4。	●削除
12:	●	左の値5は右の値4よりも大きいので交換する。	●削除
13:	●	●終端に到達したので次のループへ。	●削除
14:	●	左の値は3。	●削除
15:	●	右の値は2。	●削除
16:	●	左の値3は右の値2よりも大きいので交換する。	●削除
17:	●	左の値は3。	●削除
18:	●	右の値は1。	●削除
19:	●	左の値3は右の値1よりも大きいので交換する。	●削除
20:	●	左の値は3。	●削除
21:	●	右の値は4。	●削除
22:	●	左の値3は右の値4よりも小さいので交換しない。	●削除
23:	●	●終端に到達したので次のループへ。	●削除
24:	●	左の値は2。	●削除
25:	●	右の値は1。	●削除
26:	●	左の値2は右の値1よりも大きいので交換する。	●削除
27:	●	左の値は2。	●削除
28:	●	右の値は3。	●削除
29:	●	左の値2は右の値3よりも小さいので交換しない。	●削除
30:	●	●終端に到達したので次のループへ。	●削除
31:	●	左の値は1。	●削除
32:	●	右の値は2。	●削除
33:	●	左の値1は右の値2よりも小さいので交換しない。	●削除
34:	●	●ソート完了。	●削除

01:	●挿入	ループ開始。	●削除
02:	●挿入	左端の位置の値は1。	●削除
03:	●挿入	右端の位置の値は18。	●削除
04:	●挿入	真ん中の位置の値は9。	●削除
05:	●挿入	●9は4と比べて大きいので左側を比較。	●削除
06:	●挿入	左端の位置の値は1。	●削除
07:	●挿入	右端の位置の値は7。	●削除
08:	●挿入	真ん中の位置の値は3。	●削除
09:	●挿入	●3は4と比べて小さいので右側を比較。	●削除
10:	●挿入	左端の位置の値は4。	●削除
11:	●挿入	右端の位置の値は7。	●削除
12:	●挿入	真ん中の位置の値は4。	●削除
13:	●挿入	●4は4と比べて等しいので終了。	●削除

図 3.6: バブルソートの 2 段階目の構成と解答

図 3.7: 二分探索の 2 段階目のシステムの構成と解答

により処理の流れをコンピュータ上での表現に落とし込む支援となると考えた。web ページの構成は図 3.8 のようになっている。抽象化を支援するための図や制御構造は別のウィンドウで開ける。

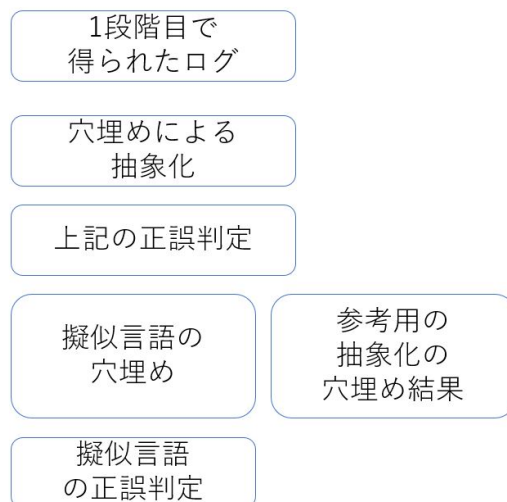


図 3.8: 3 段階目の web ページの構成

バブルソートでは図 3.9 にある変数の定義と図 3.10、3.11 による抽象化の支援となる図をもとに図 3.12 にある自然言語の選択問題を設定し、図 3.13 にある擬似言語の選択問題を設定した。図 3.9 にある「3 は array[left] で表される」、図 3.10 にある「length = 5 個」などの説明は、実際に手作業で得られたログと結びつきやすいように学習者が見る画像にも記述されている。

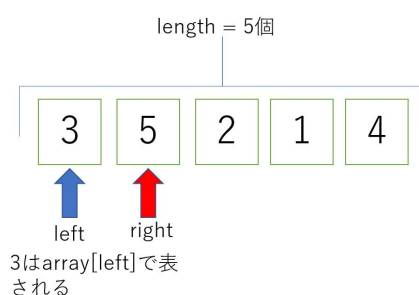


図 3.9: バブルソートの単語の定義

二分探索も同様に選択問題によって抽象化した後に擬似言語化を選択形式で行ってもらった。単語の定義は図 3.14、抽象化の支援は図 3.15、3.16、3.17、3.18 である。図 3.15 から図 3.18 まではバブルソートのときと同様に学習者には説明も含めた画像が表示される。これらの画像をみながら自然言語の選択問題である図 3.19、擬似言語の選択問題である図 3.20 に示した問題を解いてもらった。

こちらのシステムでは選択肢を変更するたびに解答にかかった秒数と正誤を保存しており、学習者は自然言語による抽象化と擬似言語化においてそれぞれ答え合わせをすることができ、間違っている問題番号を知ることができる。

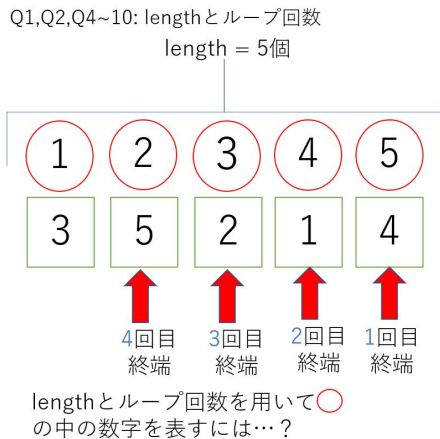


図 3.10: バブルソートの抽象化支援図 その1

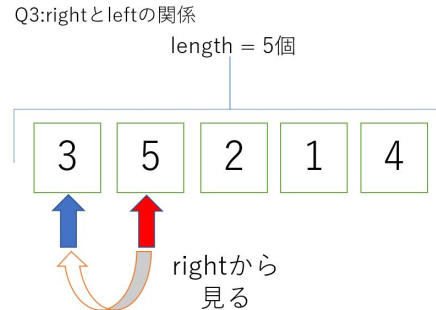


図 3.11: バブルソートの抽象化支援図 その2

```
//ループの継続条件
//まずはループの塊の数(i)による制御
for(i = (Q11) [選択待ち ▼]; i <= (Q12) [選択待ち ▼]
; i++; ){
//rightの位置によるループの塊の中の制御
for(right = [選択待ち ▼]; right <= (Q14)
[選択待ち ▼]; right++){
//値を交換する条件. 左 > 右
if(array[(Q15) [選択待ち ▼]] > array[(Q16)
[選択待ち ▼]]){
//こちらもswap(左,右)とする。
swap(array[(Q17) [選択待ち ▼]], array[(Q18)
[選択待ち ▼]]) }
}
```

ループの条件を設定するにあたり、配列の1番目から比較を行うループの塊の個数(分岐点の個数)を見る。ループの塊の個数は(Q1) [選択待ち ▼] である。これをlengthで表すと、lengthは今回5個であるため、(Q2) [選択待ち ▼] であるといえる。  
つぎにleftとrightについて見ていく。この2つは必ず隣接していることから、rightを用いてleftの位置を表すと、(Q3) [選択待ち ▼] と表すことができる。ここで各ループ内で、ループが終了するときのrightの位置を見る。1回目のループの塊では(Q4) [選択待ち ▼] で、2回目のループの塊では(Q5) [選択待ち ▼] で、3回目のループの塊では(Q6) [選択待ち ▼] となっていて最後の4回目のループでは(Q7) [選択待ち ▼] であり、ループごとに左に寄ってきている事がわかる。これをlengthとループ数を用いると(Q9) [選択待ち ▼] よりも(Q9) [選択待ち ▼] ことに気づいてほしい。  
最後に左の値と右の値を交換する条件は(Q10) [選択待ち ▼] ことである。

図 3.12: バブルソートの自然言語の選択問題

図 3.13: バブルソートの擬似言語化の選択問題



図 3.14: 二分探索の単語の定義

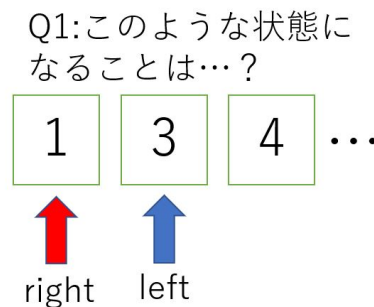


図 3.15: 二分探索の抽象化支援図その1





図 3.16: 二分探索の抽象化支援図その 2



図 3.17: 二分探索の抽象化支援図その 3



ループをするにあたり、ループを続ける条件を付ける必要がある。leftがrightよりも右側に来ることは(Q1) [選択待ち] ので、終了条件に適用できる。  
 つぎにmidからみて配列の左側、右側を比較するときの動きを考える。  
 配列の左側を比較するとき1つ目のループに該当し、leftは(Q2) [選択待ち] となる。rightは(Q3) [選択待ち] となる。配列の右側を比較するときは2つ目のループに該当し、leftは(Q4) [選択待ち] となり、rightは(Q5) [選択待ち] と表すことができる。  
 ループを終了する様子を見ると、midに該当する値であるarray[mid]が探している数字(wanted)が [選択待ち] ことである。

図 3.18: 二分探索の抽象化支援図その 4

図 3.19: 二分探索の自然言語問題

```
//初期値(ループ前に必要な値)の設定
left = 1番目
right = 9番目

//(Q1)で決まるループの継続の大小関係
while((Q7) [選択待ち] ){
    //midの算出
    mid = (left + right)/2 番目
    //midの位置の値がwanted_numよりも大きい
    if(array[mid] > wanted_num){
        (Q8) [選択待ち] = mid (Q9) [選択待ち] 1 番目
    }
    //midの位置の値がwanted_numよりも小さい
    else if(array[mid] < wanted_num){
        (Q10) [選択待ち] = mid (Q11) [選択待ち] 1 番目
    }
    //midの位置の値がwanted_numと一致した
    else{
        //何番目にいたかを渡す
        return mid 番目
    }
}
```

図 3.20: 二分探索の擬似言語問題

### 3.3 確認テスト

確認テストでは図 3.21 のバブルソートの試行回数ごとの結果を示してもらう問題、図 3.22 の二分探索の試行回数を問う問題と線形探索との試行回数の比較の問題を設定した。

1回目の終端へのループ

3	1	4	6	2	5
---	---	---	---	---	---

リセット用

解答保存

2回目の終端へのループ

3	1	4	6	2	5
---	---	---	---	---	---

リセット用

解答保存

3回目の終端へのループ

3	1	4	6	2	5
---	---	---	---	---	---

リセット用

解答保存

図 3.21: バブルソートの試行回数の状態を示す問題

16を探索する2分探索を以下の配列に対して行う。

2	4	5	6	7	8	9	12	13	15	16	17
---	---	---	---	---	---	---	----	----	----	----	----

このとき16を探索したい場合に何回目の真ん中の位置の探索で発見されるかを答えよ。ただし配列の対象範囲の要素が偶数個であった場合はアルゴリズムを習ったときと同じく、真ん中の位置は左側とすること。

参考用に以下に値を左端、右端、真ん中と選んだときに色が変わる配列を用意する。

2	4	5	6	7	8	9	12	13	15	16	17
---	---	---	---	---	---	---	----	----	----	----	----

図 3.22: 二分探索の試行回数を示す問題

### 3.4 実力テスト(挿入ソート)

実力テストでは今まで学習したことから初めて見るアルゴリズムとして挿入ソートを採用し、アルゴリズムに対して説明と図を見て処理手順を理解し、整列完了までに何回の比較が行われるかの問題、図 3.23 の1回目のループ、2回目のループ、3回目のループが終了したときの試行回数ごとの状態を示す問題、図 3.24、3.25にある自然言語、疑似言語の選択問題を解いてもらう。また、自然言語と疑似言語の選択問題については解答を選択するたびに正誤と解答時間を保存する。

付録に挿入ソートの説明の図を添付した。

1回目の末尾確認または位置決め終了時

4 1 5 2 3      リセット用

2回目の末尾確認または位置決め終了時

4 1 5 2 3      リセット用

3回目の末尾確認または位置決め終了時

4 1 5 2 3      リセット用

図 3.23: 挿入ソートの試行回数ごとの状態を示す問題

まずは整列済みの末尾確認の動きを整理する。insは最初は(Q1)  を指しており、この位置は末尾確認が終了するごとに右に移動するため末尾確認のたびに(Q2)  ことがわかる。  
 末尾確認の回数は配列の要素数をlengthとすると、insが2番目から配列の終端まで動くことから(Q3)  だけ行くと表現することができる。  
 末尾確認と位置決め動きについて整理する。  
 末尾確認のときに、何もせずに次の末尾確認に移る場合は、insが指す値(array[ins])がtargetが指す値(array[target])よりも(Q4)  場合であり、位置決めを行う条件は、array[ins]がarray[target]よりも(Q5)  場合である。こちらが位置決めを継続する条件になる。  
 また、insからみてtargetの位置は常に(Q6)  ので、targetは(Q7)  と表すことができる。  
 位置決めを行い続けると、先頭の値と比較することになりえる。先頭の値との比較を行うときにinsは配列の(Q8)  まで移動してきているため、位置決めは(Q8)の位置まで継続することになる。

```
//ループの継続条件
//for(末尾比較のはじめの状態; 終了条件(取りうる最大または最小値); 変数の増減)

for(ins = (Q9) ; ins <= (Q10) ; ins++){

//位置決定をするかの条件
//targetとinsの関係

if(array[(Q11) ] (Q12)  array[ins] ){

//今回は触れないが、insを使ってしまうと
//末尾確認に影響が出るためjを用意する。

j = ins;

//位置決定
//ループを継続する条件。
//insが最後の比較のときに移動しうる位置と
//交換、比較を継続する条件。

while(j >= (Q13)  かつ array[(Q14) ] (Q12と同じ) array[j]){
    swap(array[(Q13と同じ)], array[j])

//左への移動(Q15)
; }
}
}
```

図 3.24: 挿入ソートの自然言語の選択問題      図 3.25: 挿入ソートの擬似言語の選択問題

## 第4章 実験

### 4.1 実験スケジュールと被験者数

実験は以下のように行った。

- 内容:二分探索、バブルソートの学習後に確認テスト、挿入ソートを用いた実力テストを実施してもらった。
- 人数:研究室の修士2年1名、修士1年2名。
- 時間:二分探索、バブルソート、実力テストを10~15分、確認テストを5~10分を目処に行ってもらい、合計35分~55分実施してもらった。

対象としたかった中学3年、高校1年生では実験を行えなかったため調査内容と実験の目的を次のように設定した。

### 4.2 実験の目的と調査内容

今回実験に協力してくれた方は取り扱うアルゴリズムを学習し、プログラミングの試験も経た学生であるため実験の目的を以下の2点とした。

- わかりにくい説明や図、答えにくいUIがアンケートで調査することで適切な説明や難易度設定ができているかの調査すること。
- 解答にかかる時間を調査することで、アルゴリズムを理解している学生が解答に時間がかかる場所を特定して支援を手厚くすべき場所を見つけること。

これにより本研究の目的である学習者のアルゴリズムの処理手順の分析能力の向上、教科書による学習到達度の差をなくせるか、教師の負担を増やさない教材として十分または不十分な点はどこかを調査した。

### 4.3 評価方法

#### 4.3.1 評価の流れ

実際に機能を使ってもらうことによって収集したデータによる評価と4段階評価のアンケートを用いた。アンケートは4に近いほど高評価、1に近いほど低評価となっている。

### 4.3.2 制御構造の説明について

アルゴリズムの学習の3段階目の学習において、擬似言語化の際に不明な点がわからないように付録にまとめた制御構造の説明の図を用意したので、こちらについてもアンケートを次のように取った。

- 設問1 条件の演算子の説明は適切であったか。
- 設問2 配列の説明に用いた図はわかりやすかったか。
- 設問3 図や説明によって配列の操作はできそうか。
- 設問4 if文の図は適切であったか。
- 設問5 図や説明によってif文の穴埋めの解答に対応できるか。
- 設問6 for,while文の図は適切であったか
- 設問7 図や説明によってfor,while文を扱えそうか。
- 設問8 自由記述。

### 4.3.3 アルゴリズムの学習について

各アルゴリズムにおいて誤答した選択肢や3段階目においては解答にかかった時間を調査した。

アンケートは以下の項目を調査した。

- 設問1 1段階目のアルゴリズムは適切であったか。
- 設問2 1段階目のアルゴリズムの説明はわかりやすかったか。
- 設問3 1段階目のアルゴリズムを示す図はわかりやすかったか。
- 設問4 1段階目で総評して、手作業による経験でアルゴリズムの概要を掴めそうか。
- 設問5 図と説明も含めて、1段階目の手作業による作業で、アルゴリズムの流れは掴めそうか。
- 設問6 1段階目において配列を操作すると色が変わるようにしたが、これによって解答しやすくなったか
- 設問7 1段階目で正解の選択肢のみ受付けるようにしたが、このことで学習者は誤ったアルゴリズムを覚えないようにできそうか。
- 設問8 2段階目の学習でループごとの試行回数の変化などを理解できそうか。
- 設問9 3段階目の学習で、図はわかりやすかったか。

設問 10 3段階目の学習で、図と穴埋めの誘導に寄って体験結果とコンピュータにおける表現を結びつける(前半部分の目的)は達成することができそうか。

設問 11 3段階目の学習でコンピュータにおける表現を学習後、擬似言語化におけるコメントによる誘導は十分であったか。

設問 12 3段階目を経て最終的にアルゴリズムを理解できそうか。

設問 13 自由記述。

を調査した。

3.2.1 節で述べたように操作したボタンを記録し、3.2.3 節にあるように選択肢を変更するたびに解答と解答時間を記録した。

#### 4.3.4 確認テスト

確認テストのアンケートは4段階評価で

設問 1 問 1:バブルソートのループごとの動きの提示は問題として適切であったか。

設問 2 問 1:バブルソートのループごとの動きの提示の難易度は簡単だったか。

設問 3 問 1:バブルソートのループごとの動きの提示のUIは答えやすかったか。

設問 4 問 2:二分探索の実行回数について、問題として適切であったか。

設問 5 問 2:二分探索の実行回数について、難易度は簡単であったか。

設問 6 問 2:二分探索の実行回数について、UIは答えやすかったか。

設問 7 問 2:二分探索の実行回数について、回答前に二分探索を試行できるUIは使いやすかったか。

設問 8 問 3:二分探索と線形探索の比較について、問題の難易度は適切であったか。

設問 9 自由記述。

を調査し、確認テストの各問題の正誤を保存した。

#### 4.3.5 実力テスト(挿入ソート)について

アンケートにおいて

設問 1 問 1:挿入ソートのアルゴリズムの説明は適切であったか。

設問 2 問 1:挿入ソートのアルゴリズムの説明はわかりやすかったか。

設問 3 問 1:挿入ソートのアルゴリズムの図はわかりやすかったか。

設問 4 問 1:挿入ソートの試行回数について、難易度は適切であったか。

設問 5 問 1:挿入ソートの試行回数について、問題の難易度は簡単であったか

設問 6 問 2:挿入ソートのループごとの動きの提示について、問題は難易度は適切であったか。

設問 7 問 2:挿入ソートのループごとの動きの提示について、UIは使いやすかったか。

設問 8 問 3:挿入ソートの体験結果とコンピュータにおける表現の結びつきについて、難易度は適切であったか。

設問 9 問 3:挿入ソートの体験結果とコンピュータにおける表現の結びつきについて、難易度は簡単であったか。

設問 10 問 4:挿入ソートの擬似言語化について、難易度は適切であったか。

設問 11 問 4:挿入ソートの擬似言語化について、難易度は簡単だったか。

設問 12 問 4:挿入ソートの擬似言語化について、問題を解く上でコメントによる誘導は十分、もしくは不十分、過度であったか。

設問 13 挿入ソートを理解できるか。

設問 14 自由記述。

を調査し、実力テストの各設問に対して正誤を保存し、自然言語による抽象化と擬似言語化については選択肢を変更するたびに解答の正誤と解答にかかる時間を保存した。

#### 4.3.6 総合評価

本システムの総合評価として、

設問 1 新たなアルゴリズムを見たときに処理手順の整理して手作業で動かす力を身に着けられるか。

設問 2 総評:新たなアルゴリズムを見たときに処理手順の整理してコンピュータでの表現に落とし込めるようになるか。

設問 3 自由記述。

の質問をした。

## 第5章 実験結果

### 5.1 制御構造の説明について

アンケートの結果は表 5.1 のとおりとなった。

表 5.1: 制御構造についてのアンケート

	設問 1	設問 2	設問 3	設問 4	設問 5	設問 6	設問 7
回答者 A	4	4	4	2	4	1	3
回答者 B	4	4	4	4	4	4	4
回答者 C	3	3	4	3	4	4	4

自由記述としては回答者 A より、フローチャートの記載方法が誤っているとの指摘を受けた。

### 5.2 バブルソートの学習

#### 5.2.1 1段階目

図 3.2 において、隣り合った数字を押してから大小関係を押す流れを繰り返すことが正解となっている中で、誤った操作をした回答者はいなかった。

#### 5.2.2 2段階目

2段階目においては図 3.6 のように 13、23、30、34 行目に ● を挿入すればよいが、誤答として交換の有無を判断する 6、9、12、16、19、22、26、29、33 行目に ● を挿入する解答が 3 名から得られた。

自由記述において、● を挿入する地点はループの始点のほうがよい、何かしら動作の塊を意識させる方法があるとなおよいという意見が得られた。

#### 5.2.3 3段階目

回答者ごとの誤答した問題は最初の 1 回のみだったので、表 5.2 にまとめた。被験者が解答にかかった時間において、答え合わせを行うまでに前半の自然言語の選択問題にかかった時間を表 5.3、後半の擬似言語の選択問題は表 5.4 に示した。答え合わせのあとの



表 5.2: バブルソートの選択問題の誤答

	誤答した問題
回答者 A	Q2
回答者 B	Q1, Q2,
回答者 C	Q8, Q14, Q15, Q16

誤答修正にかかった時間は表 5.5 にまとめた。いずれも単位は秒であり、自然言語による選択問題は Q10 まで、擬似言語の選択問題は Q11 からとなっている。

表 5.3: バブルソートの自然言語の選択問題の解答所要時間

	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10
回答者 A	69	16	19	14	1	2	4	21	6	6
回答者 B	56	17	26	27	1	3	2	11	7	21
回答者 C	62	10	24	15	2	2	4	9	2	29

表 5.4: バブルソートの擬似言語の選択問題の解答所要時間

	Q11	Q12	Q13	Q14	Q15	Q16	Q17	Q18
回答者 A	25	2	2	3	4	3	4	2
回答者 B	38	1	3	13	8	2	8	1
回答者 C	16	2	3	2	3	2	3	2

## 5.2.4 アンケートの結果

バブルソートのアンケートの設問は 4.3.3 節で述べたとおりであり、結果は表 5.6、5.7 にまとめた。

## 5.3 二分探索の学習

### 5.3.1 1段階目

図 3.4 において、数字が表示されているボタンで左端、右端、真ん中に該当するものを順番に押し、「大きいので、小さいので」の大小関係を押したあとに「左側を比較、右側を比較」という一連の動作を大小関係において「等しいので終了」を押すまで繰り返してもらうことを正解とした。

その中で見かけられた誤答は、左端と右端を指定する前に真ん中に該当するボタンを押すことであった。

表 5.5: バブルソートの選択問題の再解答時間

	Q1	Q2	Q8	Q14	Q15	Q16
回答者 A		4				
回答者 B	4	3				
回答者 C			23	34	20	1

表 5.6: バブルソートのアンケート結果 1

	設問 1	設問 2	設問 3	設問 4	設問 5	設問 6	設問 7	設問 8
回答者 A	4	3	4	3	4	4	4	2
回答者 B	4	3	3	3	3	3	3	3
回答者 C	3	4	4	4	3	4	2	4

### 5.3.2 2段階目

二分探索においては、図 3.7 に示した 5、9、13 行目に ● を最初から挿入していたため誤答はなかった。

### 5.3.3 3段階目

こちらでも回答者の誤答は最初の 1 回のみだったので表 5.8 に誤答した問題、表 5.9 に 1 回目の自然言語の選択問題の解答に要した時間、表 5.10 に 1 回目の擬似言語の選択問題の解答に要した時間、表 5.11 に再解答に要した時間をまとめた。単位はすべて秒である。

### 5.3.4 アンケートの結果

アンケートの設問は 4.3.3 節で述べたとおりである。結果は表 5.12、5.13 にまとめた。

## 5.4 確認テスト

### 5.4.1 問 1:バブルソートの状態を示す問題

こちらの問題は 3 人とも正解した。  
解答に要した時間は表 5.14 のとおりである。

### 5.4.2 問 2:二分探索の実行回数

こちらでも 3 人とも正解していたため、アンケートにて問題が適切かと難易度、答えやすさについて調査した。

表 5.7: バブルソートのアンケート結果 2

	設問 9	設問 10	設問 11	設問 12
回答者 A	4	3	3	4
回答者 B	3	3	3	3
回答者 C	4	4	2	4

表 5.8: 二分探索の選択問題の誤答

	誤答した問題
回答者 A	Q8, Q9
回答者 B	Q2
回答者 C	Q1, Q7

表 5.9: 二分探索の自然言語の選択問題の解答時間

	Q1	Q2	Q3	Q4	Q5	Q6
回答者 A	17	14	2	3	3	36
回答者 B	93	7	2	6	3	3
回答者 C	17	38	2	6	2	10

表 5.10: 二分探索の疑似言語の選択問題の解答時間

	Q7	Q8	Q9	Q10	Q11
回答者 A	9	23	11	2	1
回答者 B	51	21	1	5	1
回答者 C	16	25	14	9	3

表 5.11: 二分探索の選択問題の誤答の再解答に要した時間

	Q1	Q2	Q7	Q8	Q9
回答者 A				8	8
回答者 B		28			
回答者 C	9		13		

表 5.12: 二分探索のアンケート結果 1

	設問 1	設問 2	設問 3	設問 4	設問 5	設問 6	設問 7	設問 8
回答者 A	4	3	4	3	4	4	4	2
回答者 B	4	3	4	3	3	3	4	3
回答者 C	4	4	4	4	4	4	4	3

表 5.13: 二分探索のアンケート結果 2

	設問 9	設問 10	設問 11	設問 12
回答者 A	4	4	4	4
回答者 B	3	3	3	3
回答者 C	4	4	2	4

表 5.14: 確認テスト 問 1 の解答所要時間

	解答時間
回答者 A	25
回答者 B	27
回答者 C	32

### 5.4.3 問 3:二分探索と線形探索の比較の選択問題

自然言語による文章に選択肢を用意し、解答する形にしたところ、3人共正解した。

### 5.4.4 アンケートの結果

アンケートの設問は 4.3.4 で述べたとおりである。結果を表 5.15 に示す。

表 5.15: 確認テストのアンケート結果

	設問 1	設問 2	設問 3	設問 4	設問 5	設問 6	設問 7	設問 8
回答者 A	4	3	2	3	2	4	4	4
回答者 B	4	4	2	3	3	3	3	4
回答者 C	4	3	4	4	3	3	3	3

## 5.5 実力テスト (挿入ソート)

### 5.5.1 問 1:挿入ソートの試行回数を問う問題

こちらの問題は 3人共正解した。

### 5.5.2 問 2:挿入ソートの段階ごとの状態を示す問題

こちらの問題も全員正解した。解答所要時間は表 5.16 に示す。

表 5.16: 実力テスト 問 2 の解答所要時間

	解答時間
回答者 A	20
回答者 B	24
回答者 C	15

### 5.5.3 問 3, 問 4: 挿入ソートの自然言語及び疑似言語の選択問題

回答者ごとの誤答および解答に迷った結果、正解の選択肢を選べた問題を「迷った問題」とし、表 5.17 に示した。迷った問題は、解答を変更するたびに解答所要時間と解答に選んだ選択肢が保存されるため推定できる。

自然言語の選択問題の解答所要時間は表 5.18、疑似言語による解答所要時間は表 5.19 に示した。

表 5.17: 実力テストの選択問題で誤答および迷いが見られた問題

	誤答した問題	迷った問題
回答者 A	Q10	Q12, Q13
回答者 B	Q13, Q15	Q1, Q4, Q7, Q10, Q12
回答者 C	Q8, Q10	Q11, Q12, Q13

表 5.18: 実力テストの自然言語による選択問題の解答所要時間

	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8
回答者 A	35	5	9	24	3	10	28	10
回答者 B	50	6	12	26	2	3	61	9
回答者 C	18	4	8	22	23	3	12	11

### 5.5.4 アンケートの結果

アンケートの設問は 4.3.5 節で設定したとおりである。解答は表 5.20、5.21 にまとめた。

自由記述内で、問題 1 と問題 2 の順番が逆のほうがよいとの指摘を受けた。

## 5.6 システム総評のアンケート

4.3.6 節で設定したアンケートの結果を表 5.22 に示す。自由記述では、学習の 1 段階目、3 段階目、挿入ソートの説明のときに別のウィンドウで画像を開く形にせず、説明と画像が同時に表示されるとよいとの指摘を受けた。

表 5.19: 実力テストの疑似言語による選択問題の解答所要時間

	Q9	Q10	Q11	Q12	Q13	Q14	Q15
回答者 A	10	7	1	38	83	13	5
回答者 B	7	18	12	19	10	7	12
回答者 C	21	2	21	19	13	6	47

表 5.20: 実力テストのアンケート結果 1

	設問 1	設問 2	設問 3	設問 4	設問 5	設問 6	設問 7
回答者 A	3	2	4	4	3	3	3
回答者 B	4	3	4	4	4	4	3
回答者 C	3	4	3	4	3	3	3

表 5.21: 実力テストのアンケート結果 2

	設問 8	設問 9	設問 10	設問 11	設問 12	設問 13
回答者 A	3	3	4	4	3	4
回答者 B	3	4	3	4	2	3
回答者 C	3	3	3	2	2	4

表 5.22: 総評に関するアンケート結果

	設問 1	設問 2
回答者 A	3	3
回答者 B	4	3
回答者 C	4	3

## 第6章 考察

### 6.1 段階別学習の1段階目について

表 5.6、表 5.7 の設問 4 の「図と説明も含めて、手作業による作業でアルゴリズムの流れは掴めそうか」、設問 6 「正解の選択のみを受け付けるようにしたが学習者はアルゴリズムを間違っ覚えてないか」という設問に対し、バブルソートと二分探索で共通して設問 4 には回答者 A,B,C の順に「3,3,4」、設問 6 には「4,3,4」という回答が得られ、実際に操作したログにおける誤答はバブルソートではなし、二分探索では両端の指定の前に真ん中の位置を押してしまったことのみであったという結果が得られた。そのため手作業と正解のみ入力を受け付けることでアルゴリズムを間違えずに覚えることの誘導はできていると思われる。

アンケートの設問 2 「アルゴリズムの説明はわかりやすかったか」、設問 3 の「アルゴリズムを示す図はわかりやすかったか」、設問 4 に対し、こちらも両アルゴリズムで共通の回答が得られた。設問 2 には「3,3,4」、設問 3 に対しては全員「4」との評価を受けたことから図はわかりやすかったがアルゴリズムの説明と結びつきが浅く、結果として設問 4 では設問 3 よりも「4」と回答した人数が減少したと考えられる。そのためアルゴリズムの説明と図は同時に表示することを前提に、図で動きを示している箇所のアルゴリズムの説明を囲うこと、説明の該当するブロックにマウスカーソルを置くと対応する図に枠線をつけられ、図を詳細化してスライド化やアニメーション化して説明とより結びつけられるようにすることができると思われる。

### 6.2 段階別学習の2段階目について

アンケートの結果は表 5.7、5.13 の設問 8 「2段階目の学習でループごとの試行回数の変化などを理解できそうか。」に該当し、バブルソートの方は回答者 A、B、C の順に「2、3、4」二分探索では「2、3、3」の回答が得られ、自由記述でも1段階目で得られたログに対して●を入れる地点に違和感があるという声もあった改善を行う必要がある。

実装した目的であるループの塊と内部での共通の操作や処理数を意識してもらえるように、例として二分探索では図 6.1 のように改良ができると考えた。図 6.1 においてはループの始点と終点を赤括弧で囲ってループの回数を区別し、青括弧で赤い括弧内での変数の設定などの共通の動作、分岐点は意味を変えずにもとのまま●を挿入することで青括弧で定めた内容を受けて次のループを行うか、交換が発生するか、注目する位置を変えるかといった分岐を示唆することができると思った。

01:	●挿入	ループ開始。	●削除
02:	●挿入	左端の位置の値は1。	●削除
03:	●挿入	右端の位置の値は18。	●削除
04:	●挿入	真ん中の位置の値は9。	●削除
05:	●挿入	●9は4と比べて大きいので左側を比較。	●削除
06:	●挿入	左端の位置の値は1。	●削除
07:	●挿入	右端の位置の値は7。	●削除
08:	●挿入	真ん中の位置の値は3。	●削除
09:	●挿入	●3は4と比べて小さいので右側を比較。	●削除
10:	●挿入	左端の位置の値は4。	●削除
11:	●挿入	右端の位置の値は7。	●削除
12:	●挿入	真ん中の位置の値は4。	●削除
13:	●挿入	●4は4と比べて等しいので終了。	●削除

図 6.1: 2 段階目のシステムの改善案

## 6.3 段階別学習の3段階目、実力テストの問3、問4について

### 6.3.1 誤答について

#### バブルソート

図 3.12、3.13 と表 5.2 を比較すると、バブルソートの配列の頭から末尾までの1つの塊、頭から末尾の1つ左までの1つの塊、… というようにループの塊の回数(二重ループで言うところの外側)を配列の要素数から落とし込む問題で2名、ループの塊の中で値を左右で比較して交換を進める部分の実行回数(二重ループの内側)の問題で1名が誤答している。

どちらの解答の支援をする図として図 3.15 を用意したが解答に必要な要素が混じってしまった可能性と、プログラミングに触れない教科書のレベルに合わせて配列の始点を0番目ではなく1番目としたため、被験者は普段から配列の始点を0番目とすることに慣れているためちょうど数値的に1小さい選択肢を選んでしまったのではないかと考えられる。

#### 二分探索

図 3.19、3.20 と表 5.8 を比較すると、右端の位置より右側に左端の位置が来ることはないという関係からループの継続条件を指定する問題とこれを while 文によって条件に落とし込む問題、擬似言語化において配列の左側を比較する際の変数の再設定を間違える傾向が見られた。

この結果を受けてループの継続条件に関わる図 3.15 と問題を見直すと、右端を示す矢印が最初に目に入るのに問題文では左端を主語として抽象化を行ったため誤答してしまう、つまり図から得た情報の問題文への落とし込みが注目する値によって誤答への誘導となりうることを示唆していると考えられる。

疑似言語化ではコメントによって誘導をしていたが、コメントではどの図を参照にするかを明記しなかったため疑似言語化のイメージが湧きにくくなり、アルゴリズムに慣れている被験者でも誤答があったのではないかと推測する。



## 実力テスト (挿入ソート)

図 3.24、3.25 と表 5.17 を比較すると、自然言語の選択問題によって解説を行っている Q8 を間違えていれば、Q8 を擬似言語に落とし込む問題の Q13 が誤答になる構造となると考えていたが、実際の解答では Q8 を間違えていても Q13 が正解となっていた。被験者がアルゴリズムを理解しているために起こった現象である可能性が高いが精査する必要がある。

他は配列の始点を 1 番目としているために起こったと考えられる誤答であった。

### 6.3.2 解答時間について

バブルソート、二分探索の Q1 の解答時間は資料とアルゴリズムの抽象化の支援図を読んでから解答を選択する時間を示しているがそれぞれ表 5.3、5.9、5.10 を見ると、バブルソートでは全員 60 秒を超えているが二分探索では 93 秒読んだ被験者が 1 名、17 秒で解き始めた被験者が 2 名だった。実力テスト (以下挿入ソートと呼ぶ) ではアルゴリズムの説明と図を提示して選択問題以外を解いてもらったあとに時間計測ボタンを押してから解答時間を測るがそれぞれ 35 秒、50 秒、18 秒で解き始めるというように差が出た。バブルソートは初めて本システムに触れるため抽象化の支援図や説明に目を通したため共通して 60 秒を超えたものだと推測される。二分探索では抽象化の支援図を目を通すか、挿入ソートでは問題を解く前に再度アルゴリズムの説明や図に目を通すかの差が生まれたものと考えた。

バブルソートでの解答時間を示す表 5.3、5.4、二分探索での解答時間を示す表 5.9、5.10、挿入ソートの解答時間を示す表 5.18、5.19 において回答者が 1 人でも 15 秒以上考えている問題の共通点を調査する。

自然言語による選択問題においてはバブルソートでは図 3.12 において Q2、Q3、Q4、Q8、Q10 で、二分探索では図 3.19 において Q2、Q6、挿入ソートでは図 3.24 において Q4、Q5、Q7 が該当する。これらはそれぞれのソートで定義した変数を用いて配列の要素数や変数の再定義をする場合とループを制御する条件を決定する問題であった。このことから今回の被験者が上記の思考の整理が必要な問題に時間がかかったことから、重点的に支援すべき問題は変数の再定義、制御する条件に関わる問題であることを特定できたと言える。このような問題を解く際には抽象化の支援に用いた図において定義した変数との対応を明確に表記する、思考の整理のために自分でも図をかける環境の用意が必要であると考えた。

擬似言語による選択問題ではバブルソートでは図 3.13 において Q11 で、二分探索では図 3.20 において Q7、Q8、挿入ソートでは図 3.25 において Q9、Q10、Q11、Q12、Q13、Q15 が該当する。被験者がアルゴリズムを自然言語の次の段階である擬似言語での表現に慣れているためバブルソートと二分探索では解答時間が短く、挿入ソートで該当する問題が多いのはテストであるため慎重に選択をした可能性もある。これらの共通点は for 文の初期値、while 文の終了条件の条件演算子、配列のどこの位置を指すかの変数であった。自然言語による選択問題で誘導は十分できていると考えていたが、擬似言語化の前に安達ら [20] による制御構造の図式化のシステムを用いて自然言語の選択問題で解答した選択肢がどこに入るのかなどの思考を整理するステップが必要かと思われる。

### 6.3.3 アンケートについて

アンケートの該当部分はバブルソートでは表 5.6、二分探索では表 5.13 である。挿入ソートはテストとしてのアンケートがメインであるためこちらに該当する項目は設定しなかった。

表 5.6、5.13 の設問 9「3 段階目の学習で、図はわかりやすかったか」、設問 10 は「3 段階目の学習で、図と穴埋めの誘導に寄って体験結果とコンピュータにおける表現を結びつける (前半部分の目的) は達成することができそうか」、設問 12 は「3 段階目を経て最終的にアルゴリズムを理解できそうか」である。表 5.21 のこれらの設問に 2 の評価をつけた回答者はいなかったため、自然言語の選択問題による抽象化は図もわかりやすく、アルゴリズムを最終的に理解できるシステムとなっていると言える。

一方で設問 11「3 段階目の学習でコンピュータにおける表現を学習後、擬似言語化におけるコメントによる誘導は十分であったか」の設問には 2 の評価をつけた回答者もあり、コメントによる誘導だけではなく先述の通りに自然言語の選択問題と擬似言語化の前にもう 1 ステップが必要だと考えた。

## 6.4 確認テストについて

確認テストのアンケートは表 5.15 に該当する。設問 1,4,8 がそれぞれ「問題 1,2,3 の難易度が適切であったか」であり、設問 1 は全員が 4、設問 4 は 3 が 2 名と 4 が 1 名、設問 8 は 4 が 2 名と 3 が 1 名の評価であったため、テストの問題として適切であったと言える。

設問 3「問 1:バブルソートのループごとの動きの提示の UI は答えやすかったか」の設問に対しては 2 が 2 名、4 が 1 名と改良の余地が見られた。

## 6.5 実力テストについて

実力テストのアンケートは表 5.20 と 5.21 に該当する。設問 4,6,8,11 が「問題 1,2,3,4 の難易度が適切であったか」である。この設問に対し、設問 4 は全員が 4、設問 6 は 3 が 2 名と 4 が 1 名、設問 8 は全員が 3、設問 11 が 4 が 1 名と 3 が 2 名の評価であったため、アルゴリズムを理解しているかを問う問題として適しているといえる。

## 6.6 総評のアンケートについて

表 5.22 の設問 1,2 の「新たなアルゴリズムを見たときに処理手順の整理して手作業で動かす力を身に着けられるか」、「新たなアルゴリズムを見たときに処理手順の整理してコンピュータでの表現に落とし込めるようになるか」の回答に対し設問 1 が 3 を 1 名と 4 が 2 名、設問 2 が全員 3 という評価であったため、アルゴリズムの処理手順を理解する力を身につける事ができ、コンピュータ上でのアルゴリズムの表現する助力はできるといえる。

# 第7章 終わりに

## 7.1 まとめ

本研究では教科書でアルゴリズムについて触れた高校1年生に対するアルゴリズムの学習支援教材を以下の3点の目的を満たすように設計した。

- 探索、整列アルゴリズムの処理手順を理解することを通してアルゴリズムの分析能力を身につけること。
- 上記によってアルゴリズムの取り扱いの有無に関わらず教科書による学習到達度の差をなくすこと。
- 教師の負担とならないよう、本システムで学習が完結できるように設計すること。

考察を受け、本システムは改善点が見られるものの目的を満たすアルゴリズムの学習支援の教材として有効であると判断するとともに、アルゴリズムの擬似言語化において重点的に支援すべき要素が調査できた。

## 7.2 今後の課題

今回被験者として研究室のメンバーに協力してもらったため、実際に高校1年生に対して実験を行うことがまず挙げられる。

また図や問題を著者が用意したが、教師が授業を行う際の資料として同様のことを行う場合は図と問題の整合性などの調査で時間がかかってしまうため作問や作図の支援をする研究も発展につながると思われる。

# 謝辞

本研究を進めるにあたり、日頃よりご指導いただきました中山泰一教授に厚く感謝申し上げます。

また、研究内容や発表に向けた準備等で助言などを頂いた赤澤紀子特任准教授、赤池英夫助教には大変お世話になりました。誠にありがとうございます。

中山研究室の皆様方には実験へのご参加やゼミでの助言をいただくことでご協力いただきました。本当にありがとうございました。

## 参考文献

- [1] 中山泰一:高等学校情報科の教員採用と免許外教科担任の現状, 情報教育資料, No.50, pp14-16(2020) .
- [2] 文部科学省:高等学校教員研修用教材 情報 I 第3章コンピュータとプログラミング, [https://www.mext.go.jp/content/20200722-mxt\\\_jogai02-100013300\\\_005.pdf](https://www.mext.go.jp/content/20200722-mxt\_jogai02-100013300\_005.pdf)(参照:2022/1/26)
- [3] 独立行政法人 大学入試センター:令和7年度以降の試験, [https://www.dnc.ac.jp/kyotsu/shiken\\\_jouhou/r7ikou.html](https://www.dnc.ac.jp/kyotsu/shiken\_jouhou/r7ikou.html)(参照:2022/1/26)
- [4] 独立行政法人 大学入試センター:情報関係基礎 2008 本試験 N 問題, <https://drive.google.com/drive/folders/140pQJ0KWzYH2-NvzPCyQdqFPHcZhCS0a>(参照:2022/1/26)
- [5] Algorithms Project (石田保輝, 光森裕樹):アルゴリズム図鑑, <http://algorithm.wiki/ja/app/>(参照:2022/1/26)
- [6] 長瀧寛之, 白井詩沙香:情報教育における学習支援ツールの提供・提案状況に関する現状調査, 情報処理学会研究報告, vol.2020-CE-154, No.15, pp1-5(2020) .
- [7] 恒川廉, 山田俊行:オブジェクト指向学習支援のためのメンバの可視性の視覚化, 研究報告コンピュータと教育 (CE), 2018(7), 1-7.
- [8] 相馬侑弥, 高野辰之, 小濱隆司, 宮川治:オブジェクト指向プログラミング学習支援アプリケーションの開発, 研究報告コンピュータと教育 (CE), 2017(5), 1-8.
- [9] 小清水誓太, 高野辰之, 小濱隆司, 宮川治:オブジェクト指向プログラミング教育におけるクラス図作成演習システムの開発. 研究報告教育学習支援情報システム (CLE), 2017(16), 1-8.
- [10] 鈴木優実, 松村敦, 宇陀則彦:プログラミング初学者にむけたアルゴリズム的思考習得のための学習支援手法, 研究報告コンピュータと教育 (CE), 2018, No.6, p1-7.
- [11] 安達一寿, 中尾 茂子:プログラミング学習における学生をつまづき箇所の分析, 教育情報研究, vol.10, no.4, pp11-20(1995) .
- [12] 川口順功; 情報学部. アルゴリズム構造に基づくコア・イメージのアルゴリズム教育への応用, 大学 ICT 推進協議会年次大会論文集, 2013, 6p.

- [13] 杉浦学, 松澤芳昭, 岡田健, 大岩元:アルゴリズム構築能力育成の導入教育: 実作業による概念理解に基づくアルゴリズム構築体験とその効果, 情報処理学会論文誌, 2008, 49(10), 3409-3427.
- [14] 浅野考平, 森戸隆文:アルゴリズム教育における発見学習の試み, 研究報告コンピュータと教育 (CE), 2015(21), 1-5.
- [15] 間辺広樹, 神藤健朗, 並木美太郎, 兼宗進:コンピュータ・アルゴリズムの「発見・記述・伝達」を導く授業の実践と評価, 情報処理学会論文誌教育とコンピュータ (TCE), 2(1), 10-24.
- [16] 太田剛:認知過程に考慮し, 並び替えを目指す高校情報科のアルゴリズムプログラミング教育の開発と実践, 情報教育シンポジウム論文集, 2021, 85-92.
- [17] 関下堅也, 赤池英夫, 中山泰一:探索, 整列アルゴリズム学習支援教材の提案と評価, 情報教育シンポジウム論文集, 2021, 143-146.
- [18] 独立行政法人大学入試センター:センター試験用手順記述標準言語 (DNCL) の説明 (2020.01) , <https://www.dnc.ac.jp/albums/abm00039945.pdf> (参照:2022/1/26)
- [19] Boticki. I., Barisic. A., Martin. S., Drljevic. N. : Teaching and learning computer science sorting algorithms with mobile devices: A case study. Computer Applications in Engineering Education, 21(S1), 2013, E41-E50.
- [20] 安達有希, 蜂巢吉成, 吉田敦, 桑原寛明, 阿草清滋:プログラミング学習における構文図式を用いた構文理解支援方法の提案, 情報教育シンポジウム論文集, 2018(17), 118-125.

# 付録A 1段階目のバブルソートの説明と用いた図

バブルソートは配列に対して

- ・ 1,2番目を比較。
- >1番目の値が2番目の値よりも大きければ交換。小さければ何もせず次の比較へ。
- ・ 2,3番目を比較。
- >2番目の値が3番目の値よりも大きければ交換。小さければ何もせず次の比較へ。

という操作を(最後から1つ左,最後の位置)の組まで行う。すると最後の位置には最も大きな値が挿入されるので次は

- ・ 1,2番目を比較、交換。
- ・ 2,3番目を比較、交換。
- ...
- ・ 最後から2つ左,最後から1つ左の位置の値を比較、交換。

を行う。この動作を行うと配列の最後尾の位置には最も大きな値が入るため、

- ・ 次は1~最後尾から1つ左まで。最後尾から1つ左の位置には2番目に大きな値が入る。
- ・ その次は1~最後尾から2つ左まで。2つ左には3番目に大きな値が入る。
- ...

というように最終的には1,2番目の比較、交換を行って最小値と、最小値の次に大きな値の順番を決定して終了するアルゴリズムである。

図 .1: バブルソートの説明



図 .2: バブルソートの説明図 1



図 .3: バブルソートの説明図 2



図 .4: バブルソートの説明図 3







# 付録C 実力テストの挿入ソートの説明と用いた図

挿入ソートとは、配列の先頭要素を「ソート済である」と仮定して、

・挿入位置を調べたい2番目の位置の値をソート済の部分の末尾の値(最初は1番目の要素)と比較する。(これを「末尾確認」と呼ぶ。)

・2番目の値が先頭の値よりも大きい。

->ソート済みの部分に変更はないので次の末尾確認へ。

・2番目の値が小さい。

->挿入位置を探す。(以下の比較、交換を「位置決め」と呼ぶ。) ->1番目の値と交換する。先頭まで比較したので終了。

・何もしない、交換を行ったのいずれにせよ整列済の要素が1つ増える。

・ソート済部分が1つ増えたので末尾確認を行う。

・挿入位置を調べたい3番目の位置の値を、整列済みの2番目の位置と比較する末尾確認を行う。

・3番目の位置の値が2番目の位置の値よりも大きい。

->ソート済の部分に変更はないので次の末尾確認へ。

・3番目の位置の値の方が2番目の位置の値よりも小さい。

->位置決めを行う。->3番目と2番目の位置の値を交換する。

・交換後の2番目の位置の値を、整列済みの1番目の位置と比較する。

・2番目の位置の値が1番目の位置の値よりも大きい。

->ソート済みの部分に変更がなくなるため終了。

・2番目の位置の値の方が1番目の位置の値よりも小さい。

->2番目と1番目の位置の値を交換する。先頭まで比較したので終了。

・この手順を「最後から1つ左まで整列済み」の配列の間に「最後の位置の値がどこに挿入されるか」まで行うことでソートが完了する。

図 .1: 挿入ソートの説明

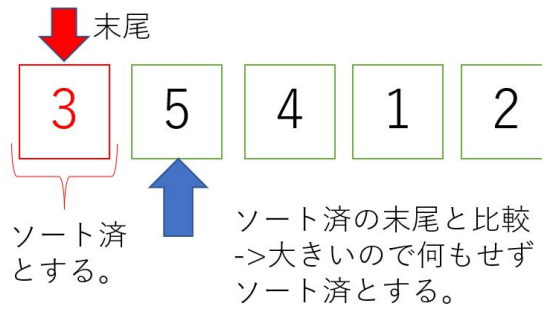


図 .2: 挿入ソートの説明図 1

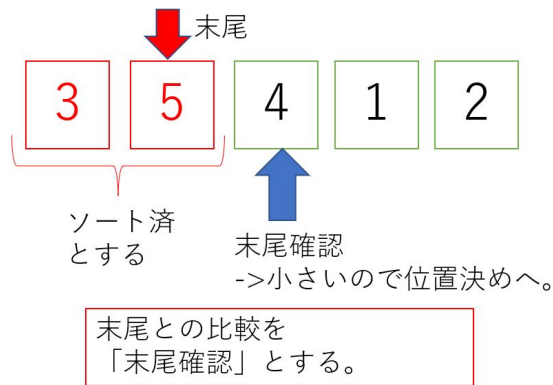


図 .3: 挿入ソートの説明図 2

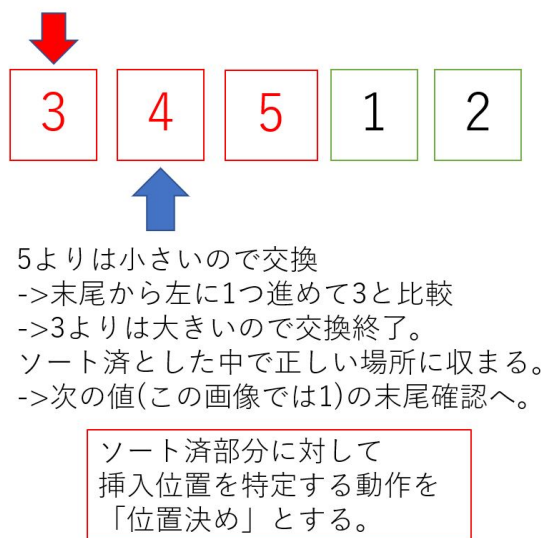


図 .4: 挿入ソートの説明図 3

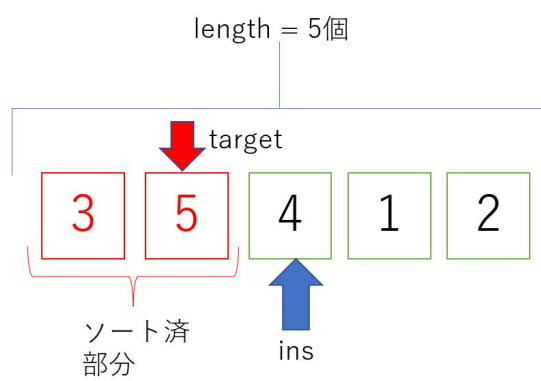


図 .5: 挿入ソートの説明図 4