

情報工学実験II  
ソーティングプログラム

イマム カイリ ルビス

令和5年6月5日

## 目 次

<b>1</b>	<b>概要</b>	<b>2</b>
<b>2</b>	<b>バケットソート</b>	<b>2</b>
2.1	プログラム . . . . .	2
2.2	動作 . . . . .	3
2.3	timecomplexity . . . . .	3

## 1 概要

ソートアルゴリズムとは、データの要素をある順序に並べるアルゴリズムである。最も頻繁に使用される順序は、数値順と辞書順で、昇順または降順のどちらかである。効率的なソートは、入力データがソートされたデータであることを必要とする他のアルゴリズム（検索やマージアルゴリズムなど）の効率を最適化するために重要である。また、人間が読みやすい出力を作成したりする際にもよく使われる。

アルゴリズムを実行するのにかかるコンピュータの時間を時間計算量 (*time complexity*) と呼ばれる。時間計算量は、一般的に *big O notation*(オーダー) という書き方で記す。例えば、 $O(n)$ ,  $O(n \log n)$ ,  $O(2^n)$ , など。

効率的なアルゴリズムとは、与えられた課題を最も早く解決するアルゴリズムであると考えることができる。そこで、この実験では、データを数値順に並べるいくつかのソートアルゴリズムを比較し、ある種の数値データに対して、どのアルゴリズムが最も効率的かを調べる。

この実験では、すべてのアルゴリズムがC言語で記述されます。ショータニング対象データは、表(??)に示すように、異なる特性を持っている。

表 1: ソータニング対象データ

データ	特徴
データ 1-3	乱数
データ 4	昇順
データ 5	降順
データ 6	バイトニック
データ 7	ジグザグ
データ 8	ランダムマイナス

spek pc

## 2 バケットソート

バケットソートは、ソートされていない配列要素をバケットと呼ばれるいくつかのグループに分割するソートアルゴリズムです。各バケットは、適切なソートアルゴリズムを使用するか、同じバケットアルゴリズムを再帰的に適用することによってソートされます。今回は、入力されたデータを、その値を配列のインデックスとして、他の配列に格納することにします。そのため、データの個数を数える必要はなく、データの最大値を知ることができます。

### 2.1 プログラム

CODE

## 2.2 動作

To make this algorithm also works for negative number, we have to know what is the max and min value of the data. So that we can set the size of bucket array will be  $(\max - \min) + 1$  ?. mainly the process of this bucket sort program is: set the bucket array to the size of max-min (+1) set the output array to the size of ammount of data iterate all the data store data at index  $(x - \min)$  bucket array store the data to output array as index + min

## 2.3 timecomplexity