

# ソート処理時間から考察する 計算量再評価

佐藤 光（東京都立日比谷高等学校 35R18） | チーム名：あせん鯽

## 研究の背景

近年、スーパーコンピュータなどを用いたシミュレーティングが科学研究をはじめさまざまな場面で活用されるようになり、その社会に貢献するところは大きい。これまでの計算量評価にはランダウによるO記法がよく使われてきた。この手法では最大次数ではない項やすべての係数などを無視するため、理論上の評価と実際の処理時間との乖離があることが知られていた。本研究においては、ソートアルゴリズムを取り上げて、その実際計算時間の集積からO記法の抱える問題に浮き彫りにし、実際との誤差を軽減する方法を検討した。

## 概要

ソートアルゴリズムを取り上げ、 $O(n^2)$ 型と $O(n \log n)$ 型のそれぞれ3種類について処理時間を計測し、O記法との差異を浮き彫りにした。その結果、傾きの差異と定数倍の誤差があることがわかった。傾きの差異は内部構造の異なりと思われるが確実ではない。定数倍の誤差は最奥のループ部分の重さに影響を受けており、逆アセンブルにより対応するバイトコードの行数を調べ、これを「ばらつきの度合い」とすることでO記法の誤差を軽減できた。

## 研究手法

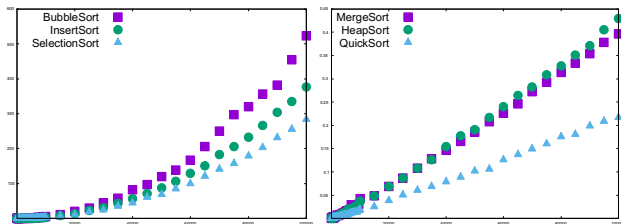
- Pythonを用いてソートアルゴリズム6種類を実装。
- 要素数(以下 $n$ とする)が100から1,000,000の昇順・降順・シャッフルの3種類の配列を用意。
- ソート処理を実行し、その実際計算時間を計測する。(timeモジュールを使用)

ソート名	最良計算量	最悪計算量	平均計算量
Bubble	$O(n)$	$O(n^2)$	$O(n^2)$
Selection	$O(n)$	$O(n^2)$	$O(n^2)$
Insert	$O(n)$	$O(n^2)$	$O(n^2)$
Heap	$O(n \log n)$	$O(n \log n)$	$O(n \log n)$
Merge	$O(n \log n)$	$O(n \log n)$	$O(n \log n)$
Quick	$O(n)$	$O(n^2)$	$O(n \log n)$

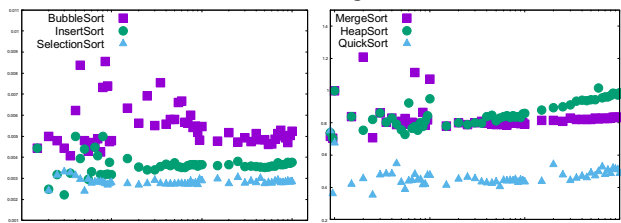
出典: 明治大学『並べ替えのアルゴリズム』(QRコード参照)

## 実験結果

左が $O(n^2)$ で表される3種類のアルゴリズムの処理時間であり、右が $O(n \log n)$ で表される3種類のアルゴリズムの処理時間を示したグラフ。



ここで、これらの縦軸をそれぞれ $n^2$ と $n \log n$ で除したものが下のグラフ。



## 問題点の分析

上に示したグラフのうち、下段の2つをみると処理時間にはそれぞれ定数倍の誤差が生じていることがわかった。これはプログラム中の処理の重さの違いによって生じているだろうという仮説を立てた。

また、特に $O(n \log n)$ 型におけるHeapソートで顕著であるが、定数倍の誤差のほかに傾きの差異も生じていることがわかる。なお、これらの $O(n \log n)$ 型ソートについて、MergeソートやQuickソートはループを繰り返しながら配列を分割していく操作であるのに対して、Heapソートはより複雑な機構をしている (QRコード参照)。この構造の違いが傾きの差異を生んでいると予想されるが、実証する手立ては掴めていない。

以降の操作では、定数倍の誤差を軽減するための方法を検討した。

## 修正の検証

### ● 逆アセンブルとは

プログラミング言語を機械が処理できる機械語に翻訳する作業をアセンブルと呼ぶのに対して、機械語の可読性を上げる操作のことを逆アセンブルと呼ぶ。本研究ではバイトコードの解読のために用いた。

### ● 検証内容

- 実験に用いた6種のソート関数を逆アセンブルする。
- 最奥のループに対応するバイトコードの行数を調べ、その行数を $W$ とする。
- 本ポスター左側下段のグラフの縦軸を $W$ で除したものを $B$ とする。
- その数値の(標準偏差)/(平均値)を「ばらつきの度合い」とする。

以上の操作によって評価の不正確さが軽減されたかどうかを検証した。以下に「ばらつきの度合い」を示した表を記す。

Algorithm	$W$	$B$	$B/W$	Algorithm	$W$	$B$	$B/W$
Bubble	最良 22	27.2686693	1.23948497	Merge	81	250.0034537	3.086462391
	最悪 48	74.4970717	1.55202233	Quick	35	138.9740454	3.970687013
Insertion	最良 48	72.8952761	1.51865159				
	最悪 22	27.1473644	1.23397111				
Selection	最良 25	29.1457744	1.16582978				
	最悪 25	29.1457744	1.16582978				
平均値		46.1908252	1.34199195	平均値		194.4887496	3.528574702
標準偏差		25.1277111	0.17925588	標準偏差		78.50964746	0.625241226
標準偏差/平均値		0.54299788	0.13357448	標準偏差/平均値		0.403671923	0.177193705

この表から、 $W$ で除したほうが「ばらつきの度合い」が低下しており、 $W$ を用いて相対評価することで、定数倍の誤差を軽減させることができるとわかる。

## まとめ

これら一連の研究を通して以下のことが結論づけられる。

- O記法による計算量評価はたしかに実際の計算時間と差異があり、それは定数倍の誤差と傾きの差異であった。
- 傾きの差異が生じている関数同士では、再起などの内部構造に大きな違いが見られるが、結論には至っていない。
- 定数倍の誤差について、関数を逆アセンブルし、ループの最奥に対応するバイトコードの行数がこれと深く関わっていた。

## 今後の展望

本研究は特に定数倍の誤差について着目しているが、インタプリタであるPythonに限定しているため、コンパイラ言語であるプログラミング言語Cなどでは挙動が異なることが予測される。同様のことをCにおいても検証したが、有意な結果が得られなかった。また、傾きの差異についての考察は $O(n \log n)$ 型のHeapソートのみとなっており、性質を決定するには不十分であるといえる。今後の研究によりデータが積み上がれば、コンピュータを用いた計算やシミュレーティングがますます浸透するだろう。

本研究の論文です (本文10 ページのPDF)  
より詳細な説明と厳密な議論を記述しています。  
GitHubページに移ります。

