

多重分割ソートの有効性とその改良について

4 B-5

○吉岡正史 常念一知 志田晃一郎 藤川英司 山田新一
武蔵工業大学工学部

1 はじめに

クイックソートは平均的に最速のソート法として知られているが、1994年、これをある条件下で上回るソート法として多重分割ソート (Multi Partition Sort: 以下 mps) が河村らによって発表された^[1]。原著者らによる mps アルゴリズムは、ソート対象の配列 A の要素移動回数を抑えるために非常に複雑なものとなっている。しかし、mps は要素サイズの大きい配列に対して有効にソートするという特徴を持つ。

そこで、本研究では mps のアルゴリズムを整理した改良型を開発した。このアルゴリズムは、mps の分割数 b と区間数の不一致を解消し、最適な要素の移動方法を実現させている。本稿では、mps のふるまいに注目しつつ、この改良型の内容と mps の正しい評価について報告する。

2 mps の原理と問題点

クイックソートは、ソートする配列より基準となる要素を一つ取り出し、その値をもとに大小2つの要素群に分割することを再帰的に繰り返して整列を完了させる^[2]。mps では、一度に多数の区間へと分割し、比較的近い値の集まったそれぞれの区間をソートさせて整列を完成させる。分割数は2のべき乗にすると効率的である。

mps では、この分割作業を行うために多数の作業領域を必要とする。mps の改良は、ソート効率を低下させずに作業領域を減らし、複雑さを軽減していく方向が望ましい。

又、mps の用いる作業領域のうち最大のものは

要素数 n に対しオーダー n となるが、この領域を許すのであれば、要素の入れ替えを行わずにポインタを継ぎ替えて整列させることも可能である。この方法はアドレステーブルソーティング (Address Table Sorting: 以下 ATS) と呼ばれる^[3]。mps の有効性を確認するために、ATS との比較、検討を行う必要がある。

3 mps の改良

3.1 分割数と区間数の一致

mps は、分割の基準値自体も独立した区間として取り扱うため、実際に分割される区間数は $2b-1$ で分割数 b と一致しない。そこで、この不一致を解消して実装を行った^[4]。

3.2 要素移動の改良

河村型 mps の要素移動の様子を図1に、今回改良を行ったものを図2に示す。河村型の移動方法は、配列 A とは別に長さ $2b-2$ (分割数を一致させたモデルは $b-1$) のスタック T を用意する必要がある。前回^[4]、このスタックをなくして作業変数を1つ使用する移動方法を発表し、河村型の97.8%の実行時間を得たが、この方法はまだ効率的なものとはなっていなかった。

今回開発したアルゴリズムは、作業変数を2つ用いて移動を行う。この方法を用いることによって、配列 A については同じ移動回数、全体の計算量についてはより少ない回数で整列が完了する。実際の実行時間、代入効率については当日発表する。

4 ATS を用いたクイックソートとの比較

A study on Multi Partition Sort
Masafumi Yoshioka, Kazuchi Jonen, Koichiro Shida,
Hideji Fujikawa, and Shin'ichi Yamada
Faculty of Engineering,
Musashi Institute of Technology

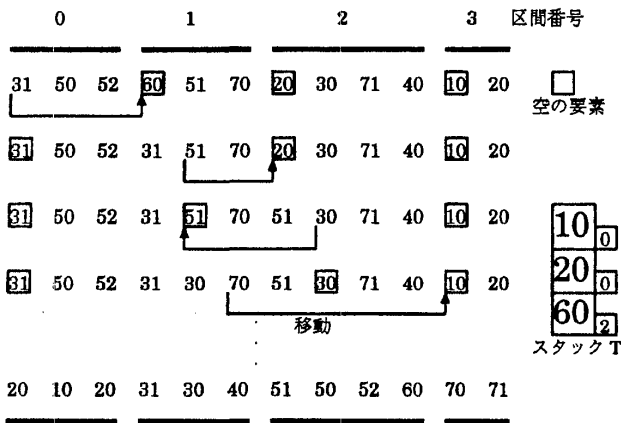


図 1: 河村型の要素移動の様子

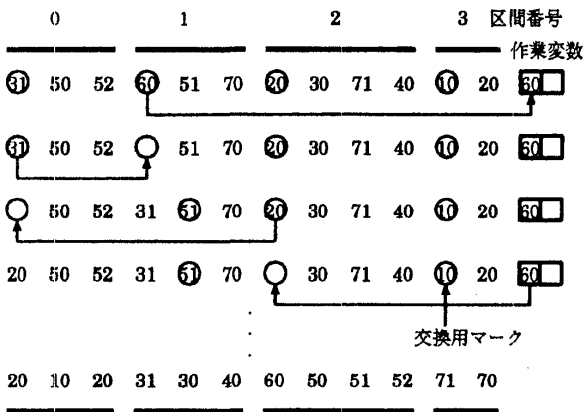


図 2: 改良した要素移動の様子

図 3 に示す ATS は配列を直接移動するのではなく、各要素に割り振ったポインタ列をソートして、最終的にポインタを先頭から参照して整列状態を得るテクニックである。次に、このポインタ列を元に配列 A のソートを行うが、すでにソートのガイドラインができているためにこの作業は必ず n 回の要素移動で完了する。この方法はどんなソートアルゴリズムにも適用することができる。

サイズの大きな配列に対して mps が有効であるのであれば、ATS を用いたクイックソートよりも

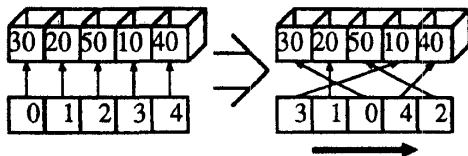


図 3: ATS の様子

高速であるか否か確認する必要がある。

ATS を基準とした mps の性能を表 1 に示す。ここで示す mps はまだ要素移動の改良を施していないものであり、値は分母をクイックソートとした実行時間比である。これより、現段階においては大きな要素に対して mps が ATS クイックソートより高速にはなっていないことがわかる。要素移動を最適にした mps との比較の結果については発表当日に示す。

表 1: mps と ATS クイックソートの実行時間比

要素サイズ M [bytes]	クイックソートの種類	
	ATS なし	ATS あり
20	101.7	113.2
40	97.8	108.7
60	96.8	108.0
80	95.5	107.8

% = mps/qsort

5 結論

mps は、配列 A とは別に多くの作業領域を用いて要素同志の少ない移動回数を実現している。mps の評価を正確に行うには、河村らの複雑な実装を単純化してアルゴリズムの骨格を明確にする必要がある。これと ATS を比較することで、mps が本当に有効な方法かどうかが明らかになる。

参考文献

- [1] 河村, 江口, 小笠原, 重村: 多重分割ソートアルゴリズム, 情報処理学会論文誌, Vol.35, pp.2537-2544 (1994).
- [2] 奥村 晴彦: C 言語による最新アルゴリズム辞典, 技術評論社 (1991).
- [3] D. E. Knuth: The Art of Computer Programming, Vol.3, pp.74-75, Addison-Wesley (1973).
- [4] 吉岡, 志田, 藤川, 山田: 多重分割ソートのふるまいの研究とその改良, 情報処理学会第 52 回全国大会講演論文集 (1), pp.63-64, (1996).