Vol.2020-CE-156 No.1
2020/8/29

コンピューターの整列処理におけるデータ操作の時間的共起分析

山口 琢[1]　大場 みち子[2]

概要：学習アプリケーションをうまく設計すれば、学習者の考え方が操作のパターンに表れると、我々は期待している。例えば、並べ替えプログラミングの操作を測定・分析して、正解・不正解にかかわらず学習者の解き方を推定してきた。では、並べ替えのアルゴリズムがあらかじめ分かっているとき、外から観察できる並べ替え操作を測定・分析して、結果の違いをアルゴリズムの違いで説明できるだろうか？そこで、1 から 10 の 10 個の数を、よく知られた整列アルゴリズムでコンピュータに整列させ、途中の数の交換・移動といったデータ操作を記録した。交換操作は、添字の大きな数を小さな方の前に移動し、互いに隣り合っていれば操作終了、隣り合っていなければ添字の小さな方を大きな方の元の位置に移動するという順序で記録した。操作対象に着目して時間的な共起を分析したところ、ヒープソートではその数より 1 つ大きい/小さい数と前後して動かされることが多く、バブル・ソートではその数自身と前後して動かされることが多いなど、整列アルゴリズムの特徴が共起行列に表れた。コンピュータによる整列はどれも正解するが、内部的なアルゴリズムが異なると、外から観察できる並べ替え操作が異なることを確認できた。

Temporal Cooccurrence Analysis of The Data Operations in The Sorting Processes by Computers

Taku Yamaguchi[1]　Michiko Oba[2]

1. 背景

特にプログラミング教育において、学習者の演習プロセスを測定・分析して、学習者のつまづきや理解を測ろうという研究が行われるようになってきた [1][2]。

学習アプリケーションをうまく設計すれば、学習者の考え方が操作のパターンに表れると、我々は考えている。プロセスを分析することで、これまで認識されていない未知の・言語化されていない思考力・考え方を捉えられると期待している。

1.1 人による並べ替え操作の測定・分析

我々はこれまで、並べ替えプログラミング・パズルのジグソー・コードを開発して、学習者が並べ替えプログラミングするとき、並べ替え操作を測定した。操作の傾向を分析して、学習者へアドバイスすべき内容を明らかにした [3][4]。

我々はまた、人の並べ替え操作といった非連続的な時系列データを分析する手法として、時間的な共起分析を提案してきた。これは、2 つの操作が時間的/順序的に近くで頻繁に起きる (共起する) なら、それら操作の間に何らかの関係があるだろうとの考えのもとに、共起頻度を分析するものである [5]。

1.2 コンピュータによる整列

コンピューターによる整列はコンピューターの歴史と共にある。従来は性能、すなわち所要時間 (所要ステップ数) や必要なメモリの観点、あるいは安定性、すなわち同じソートキーの要素が複数あたっとき、整列によって元々の順序が保たれるかどうかの観点で評価されてきた [6]。

[1] フリー
Independent Researcher
[2] 公立はこだて未来大学システム情報科学部
Faculty of Systems Information Science, Future University Hakodate

ⓒ 2020 Information Processing Society of Japan

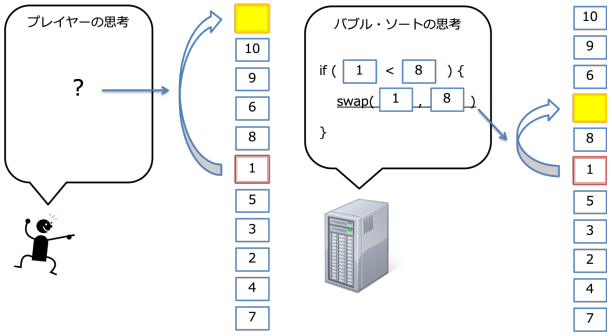

図 1　コンピューターによる並べ替えの測定・分析

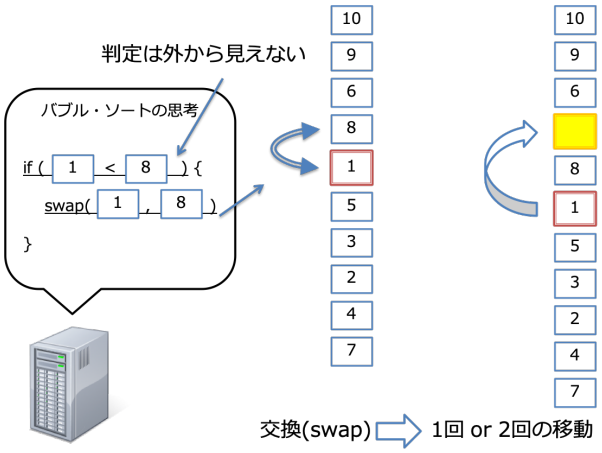

図 2　コンピューターによる整列操作の測定

コンピュータによる整列 (並べ替え、ソーティング) は、本稿の観点から次の特徴を持つ:

- 必ず正解する。正答率の変化で評価するとか正解を教師として機械学習するといった、学習分析でよく採用される手法を使えない。
- アルゴリズムが明らかである。本人にインタビューして「何考えてましたか？」と聞くまでもない。何より自分で組んだアルゴリズムで動いているのだから。
- 再現できる。人間による整列よりもコンピューターによる整列の方が再現性が高い。
- 網羅性が高い。人間に並べ替え可能な 10 個程度の数であれば、パソコンで全順列を網羅できる。

機械学習では、コンピューターに解かせる問題として整列を取り上げることがある。我々のアプローチ同様に、整列のプロセス (文献では traces と呼ばれている) を教師として学習させる研究もある [7]。

2. 目的

本稿の基本的な問は、コンピュータによる整列を外から測定・分析して、分析結果の違いをアルゴリズムによる違いで定量的に評価できるか？である (**図 1**)。整列アルゴリズムを可視化して見られる Web サイトが数多くある。これらは、整列アルゴリズムを指定して、並べ替える様子を可視化するものである。本稿は逆に、外から観察できる並べ替えの様子から、その内部のアルゴリズムに迫るものである。

ここで、整列アルゴリズムにおける「比較」処理は外から観察できないとして、記録の対象外とする。

3. アプローチ

1 から 10 の 10 個の数を整列アルゴリズム (整列法) でコンピュータに整列させ、途中の数の交換・移動といったデータ操作を記録する。記録した操作を時間的共起分析 [5] で分析して、結果の違いを、アルゴリズムの違いで説明するよう試みる。

3.1 作問

1 から 10 の 10 個の数を昇順に並べ替えることを問題とする。全順列 3,628,800 個の問題をコンピュータに解かせる。この中には並べ替え不要のものも含まれる。各問題は、数のリストとして各ソート処理に入力される。

3.2 操作の測定

将来、人間による並べ替えと比較することを想定して、ソート・アルゴリズムにおける数の「交換」を、数の「1 回または 2 回の移動」に翻訳して記録する**図 2**。また、交換のきっかけとなる数の比較/判定は、外部からは見えないので測定・分析の対象外とする図 2。

今回取り上げるソート・アルゴリズムはどれも、リストを走査しながらある条件が満たされると 2 つの数を交換 (swap) する [6]。ジグソー・テキストなど、ドラッグ&ドロップで並べ替えするアプリケーションには交換する操作はない。そこで、ジグソー・テキストに合わせて、ソート・アルゴリズムの「交換」操作を次のような操作とみなす:

(1) まず、問題リストにおける添字の大きい方の要素を移動、すなわちドラッグ&ドロップしたとみなす。
(2) 両者が隣り合っていたら、これで移動は完了。
(3) 隣り合っていなかったら、続いて添字の小さい方を移動したとみなす。

3.3 クイックソート

クイックソートでは、分け目 (ピボット) の取り方が影響する。今回は、2 つの要素の添字の、真ん中の添字の要素を分け目とする。

3.4 処理の流れ

コンピューターによる並べ替え操作を測定する処理の流れは次の通りである:

(1) 1 から 10 の 10 個の数字の全順列を生成してパズル群とする。

ⓒ 2020 Information Processing Society of Japan

n \ n+1	1	2	3	4	5	6	7	8	9	10
1	13,063,680	595,200	670,880	616,896	507,360	382,848	263,520	158,400	70,560	0
2	1,128,960	10,160,640	661,178	709,932	646,340	515,808	365,760	222,720	99,360	0
3	529,200	1,411,200	7,620,480	910,176	737,500	616,288	448,416	276,384	123,600	0
4	272,880	730,800	1,411,200	5,443,200	1,286,880	684,688	505,368	314,664	141,312	0
5	154,920	395,280	806,400	1,209,600	3,628,800	1,645,168	524,316	328,216	148,644	0
6	92,784	224,208	464,352	745,920	887,040	2,177,280	1,791,420	300,068	138,728	0
7	55,758	129,192	267,408	446,400	569,520	524,160	1,088,640	1,574,748	98,378	0
8	31,310	70,758	146,136	249,600	331,920	327,600	201,600	362,880	953,498	0
9	13,699	30,714	63,660	110,400	150,120	152,880	100,800	0	0	0
10	0	0	0	0	0	0	0	0	0	0

図 3 バブル・ソートの共起行列
セルの値の合計 (=全操作数) 78,019,201; 平均 780,192; 標準偏差 1,868,693.9

(2) (3)〜(5) を全順列について繰り返す。
(3) パズルをバブル・ソートで並べ替え、交換操作を記録する。
(4) パズルをクイックソートで並べ替え、交換操作を記録する。
(5) パズルをヒープソートで並べ替え、交換操作を記録する。

3.5 実行環境

作問 (全順列の生成)、並べ替え処理、時間的共起行列の生成は Python3、macOS Catalina、Mac mini (2018、3.6 GHz クアッドコア Intel Core i3、メモリ 8GB) で実行した。共起行列の表示、標準偏差など計算は、同マシン上の Google Chrome ブラウザで JavaScript で行った。

4. 結果

以下に、各整列アルゴリズムについて、時間的の共起行列を示す。**図 3** から**図 5** の共起行列において、背景が赤い (濃い背景色の) セルは値が平均+標準偏差 x2 よりも大きい、背景が黄色 (薄い背景色) のセルは値が平均+標準偏差より大きいことを表す。

4.1 バブル・ソート

図 3 では、1〜5、すなわち小さい方の数について、対角線上のセルの値が大きい。バブルソートではその数自身と前後して動かされることが多い。すなわち、小さい数ほど続けて動かされることが多いことを、時間的共起行列は示している。

大きな数ほど動かされることが少ない。特に 10 は動かされない。

4.2 クイックソート

図 4 は、3 つの中では最も操作数 (整列処理のステップ数) が少ない。また、極端に頻度が大きい組み合わせが、他の 2 つに比べて少なく、赤い背景 (濃い背景) のセルがない。これは、3 つの中で最も速いアルゴリズムであることと関係すると考えられる。

n \ n+1	1	2	3	4	5	6	7	8	9	10
1	533,400	404,520	664,220	709,792	627,770	569,382	521,569	430,344	377,547	334,896
2	55,512	330,792	393,940	591,884	643,412	612,126	572,400	465,422	401,321	347,040
3	191,408	154,180	323,848	428,540	628,932	711,788	687,245	544,890	455,513	368,040
4	581,762	343,798	204,750	253,388	402,562	605,533	713,512	586,496	481,105	358,200
5	420,714	410,502	341,188	212,100	209,086	391,977	626,804	651,199	531,405	359,352
6	470,138	433,990	415,608	336,266	134,819	192,840	498,395	674,131	647,392	414,012
7	483,562	426,628	462,432	439,514	273,830	157,665	234,206	573,350	683,807	466,740
8	419,436	394,330	426,916	452,942	391,489	288,112	128,844	199,219	482,283	478,560
9	410,208	370,108	391,506	425,446	401,518	407,129	257,670	87,482	105,982	217,680
10	429,644	381,696	384,936	402,712	354,246	445,546	396,119	226,613	73,238	0

図 4 クイックソートの共起行列
セルの値の合計 41,018,041; 平均 410,180.4; 標準偏差 161,786.2

n \ n+1	1	2	3	4	5	6	7	8	9	10
1	3,938,756	2,908,320	3,908,456	3,505,804	2,414,256	2,213,716	1,630,812	2,159,600	1,103,232	1,233,976
2	6,360,968	1,728,344	2,661,264	2,016,332	2,219,548	1,787,904	1,656,784	1,537,396	1,412,516	1,386,012
3	6,411,964	6,324,196	695,640	622,256	1,083,552	1,165,272	1,295,364	1,059,988	1,421,492	1,406,268
4	3,982,384	3,593,440	5,625,328	390,768	409,000	645,736	848,116	725,644	1,267,180	1,320,916
5	2,505,680	2,451,816	2,708,272	5,911,952	185,088	283,208	522,440	507,368	1,007,816	1,176,104
6	1,983,464	1,853,640	1,791,968	2,055,544	6,042,056	67,200	250,292	363,092	704,852	1,023,092
7	1,582,248	1,495,800	1,409,064	1,449,136	1,738,564	6,004,612	17,280	267,552	456,192	915,552
8	954,864	1,212,352	1,248,080	1,206,900	1,263,408	1,609,152	5,972,352	0	302,286	907,086
9	532,392	640,392	709,352	749,112	825,678	1,085,046	1,646,046	6,273,006	0	401,056
10	392,112	492,912	589,632	681,352	772,648	876,256	1,005,856	1,199,056	4,513,696	0

図 5 ヒープソートの共起行列
セルの値の合計 174,871,524; 平均 1,748,715.2; 標準偏差 1,650,555.5

4.3 ヒープソート

図 5 は、各数について、対角線の左隣のセルの値が最も大きい。すなわち、ヒープソートではその数より 1 つ大きい/小さい数と前後して動かされることが多いことを示している。

4.4 ヒストグラム

図 3 から図 5 の各共起行列のセルの値 (共起頻度) の分布を、横軸を同じにしたヒストグラムとして**図 6** に示す。横軸は数同士の時間的な共起頻度に対応し、縦軸は共起頻度の値が各区間に存在する頻度に対応する。セルの数は、どれも 10 x 10 = 100 個で、棒の値の合計は 100 である。共に頻度ということばが使われるので注意深く解釈すると、バブルソートは、共起頻度が大きい数の組が存在して横軸の右側まで分布する。かつ、そのように大きな共起頻度の数の組の数は少ない。

個別のヒストグラムは、付録の A.1 節に示す。

5. 考察

内部のアルゴリズムの違いによって外から観察できる並べ替え操作が異なる。それぞれの特徴をアルゴリズムに基づいて、以下に説明する。

5.1 バブル・ソート

バブルソートのアルゴリズムでは、隣同士を比較して小さい方を添え字の小さい方へ移動する。このバブル・アップの動きを、共起行列は可視化している。小さい数ほど対

ⓒ 2020 Information Processing Society of Japan

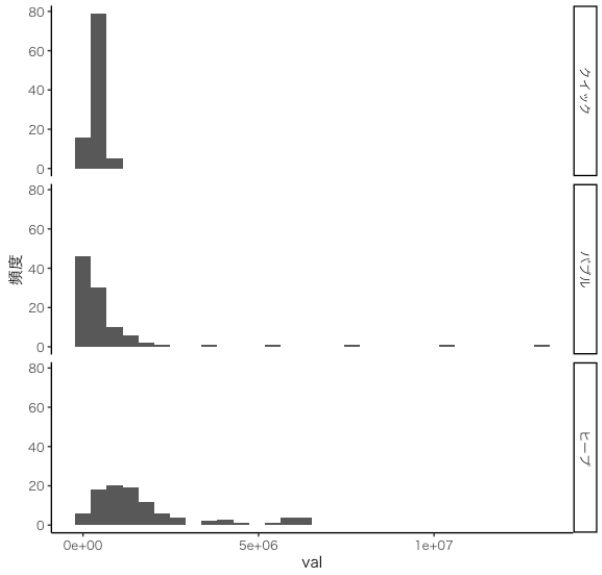

図 6　共起頻度 (共起行列のセルの値) の分布

角線上のセルの値が大きいのは、小さい数バブル・アップの操作が連続するからと考えられる。

10 のピースが動かされないのは、バブル・ソートのアルゴリズム上の特性である。バブルソートでは、隣同士を比較して小さい方を動かすので、最も大きい 10 は全く動かされない。

5.2　クイックソート

クイックソートは、そもそも操作数が少ない。クイックソートが 3 つの中で最も高速であることの表れと考えられる。

5.3　ヒープソート

ヒープソートは、ヒープを再構成しつつヒープ内の最大数を取り出す処理を繰り返す。大きい方から順にヒープから取り出されることが、対角線の左隣、すなわち自身よりも 1 つ小さい数と前後して動かされることが多いことと関係すると考えられる。図 5 の結果は、ヒープを再構成するまでもなく続けて最大値を取り出せたことが、全体を通して集計すると多かったことを示すと考えられる。

5.4　人間による整列の予想とヒープソート

ここで、人間による数の整列を測定して同様に分析した結果を予想してみる。**図 7** は、1 から 10 の 10 個の数を整列するパズルを人間が解いたときの操作の、共起行列の予想である。**図 8** は、6 桁の数を 10 個 (例えば、s2=530267、s3=533180、…) を整列するパズルについての予想である。具体的な数値は省略していて、背景色の意味は 4 章と同じである。

この背景色の分布が予想の中身である。すなわち、1 から 10 の数の場合は、1 が一番小さいとすぐに分かるので、

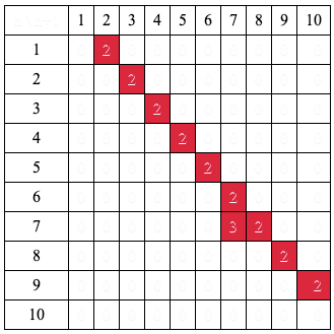

図 7　人間による 1 から 10 の 10 個の数の整列の共起行列 (予想)

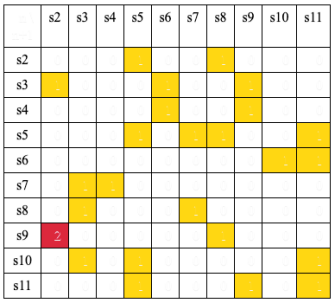

図 8　人間による 6 桁の数 10 個の整列の共起行列 (予想)

1 から昇順に並べていくだろうと予想する。その結果、1 の次に 2、2 の次に 3 と順に動かすので、対角線の 1 つ右側のセルの共起頻度が突出して高くなるだろうと予想できる (図 7)。

6 桁の数では、どれが一番小さいかは直ちには分からず、数同士を比較しながら並べ替えていくだろうと予想する。その結果、図 7 のようにはならないと予想できる (図 8)。

ヒープソートは、人間が 1 から 10 の 10 個の数を並べ替えるプロセスに近いのかもしれない。ヒープソートの操作記録には、ヒープの再構成が含まれている。ヒープとは容易に最大値や最小値を取り出せるデータ構造である。ヒープを構成し、最大値を整列済のデータに追加する、という操作を繰り返すのがヒープソートである。1 から 10 の 10 個の数であれば、人間はすぐに最小値を取り出せる。ヒープ再構成以外の操作が外に表れると考えれば、1 から 10 の 10 個の数の整列で、ヒープソートは人間による整列操作に似ると考えられる。

5.5　測定方法の影響

4 章の共起行列は、測定方法に影響される。整列処理中の数の交換について、添字の小さい方を先に記録すると、全く異なる共起行列となる。これら共起行列を付録の A.2 に示す。

今のところ、最適な測定方法を事前に知る手法は見当たらない。今回は「ヒープソートの特徴を可視化するには、添字の大きい方を先に記録する」ことで特性を効果的に検出できた。別の文脈では、違うかもしれない。測定手法や

ⓒ 2020 Information Processing Society of Japan

分析手法は手段にすぎない。測定手法や分析手法の選択肢は多いほうがよい。

5.6 分析手法

今回は、どのアルゴリズムで整列したか分かったうえで、アルゴリズムごとにデータを分けて、それぞれの記録を時間的共起行列を分析した。3 つのアルゴリズムの記録を混ぜて分類するアプローチもあるだろう。これについては 6 で検討する。

6. 残された課題

以下、残された課題を提示し、予想に予想を重ねて、本稿の意義を論じる。

6.1 人間による整列の測定・分析

人間による整列を測定・分析して、5.4 節の予想を確かめる。数の整列、特に 1 から 10 の数は間違えにく、正答/不正解と関連付けて分析するのは難しいだろう。プロセスに基づく分析が有効だろうと予想する。

6.2 コンピューターによる整列の機械的な分類

本稿のような測定データを機械的に分類したとき、その分類は整列アルゴリズムの違いと一致するだろうか。本稿は、どの整列アルゴリズムによるものか知ったうえで、各並べ替え操作の記録を分析して、アルゴリズムに基づく説明を試みた。

6.3 人間による整列の機械的な分類

人間による整列プロセスは、コンピューターによる既知の整列アルゴリズムに似ているだろうか？コンピューターによる整列の分類器を人間による整列に適用したとき、結果はどうなるだろうか。どれかに一致、いくつかの整列アルゴリズムに部分的に一致、どれとも一致しないなど予想できる。

コンピュータによる整列と人間による整列が、ある分析に基づいて似ているにしろ、似てないにしろ、その内容を具体的に確かめる。人間にやり易い整列方法、コンピューターにやり易い整列法は異なるかもしれない。

6.4 人間のトレーニング

人間向きの「6 桁の数の効率的な整列方法」があるならば、それを人間に教えられるだろうか？そのような操作の仕方に誘導する仕組みをジグソー・テキストに導入して、何 10 回、何百回と繰り返して数の整列パズルを解く。学習者が数を整列するときの操作数 (手数) や所要時間の変化で、効果を評価する。

機械学習では Curriculum Learning という手法が使われることがある。これは、学習させるときに、優しい問題から学習させる手法である [8]。このコンセプトを人間に逆輸入するなら、数の整列のトレーニングをするとき、桁の少ない数から初めて桁数を増やしたり、並べ替える数を 5 個から初めて増やしたりといった手法が該当するだろう。このように最初は優しい課題から始めるか否かで違いを評価することが考えられる。

機械学習では、なぜそう判断するのか人間が説明できないような判断を、機械に学習させられる。ジグソー・コードのような仕組みを、入学試験のような判定装置ではなく、野球の素振りのようなトレーニング装置として使う。機械学習同様に、まだ名前もついていない未知の「考え方」すらも、人間に教えられるかもしれない。

6.5 人間のふりをするコンピューター vs. 見破る分析

バブル・ソートの共起行列 (図 3) では、10 が全く動かされない。全順列 3,628,800 個のパズル問題について延べ 78,019,201 回操作して、一度も動かされないとは人間業とは考えられない。人間のふりをして整列するコンピュータと、それを見破る分析とで対戦するハッカソンなどが成立するかもしれない。

前節までを含め、このような課題に取り組むことで、人間らしさとは何かが浮かび上がると期待する。

6.6 他のパズル

本稿の手法を他のパズルに適用することで、知見がより深まると期待する。コンピューターに解けて、人間にも解けるパズルである。コンピューターでパズルを解く研究は歴史が深い。15 パズル、ハノイの塔などが考えられる。

Web アプリならばインストール不要で実験の自由度が高い。オープン・ソースならば、本稿の 3.2 節と同様に、操作を測定する仕組みを検討して追加すれば実装が完了する。

数の整列では、交換 (swap) 処理の記録の仕方によって分析結果が異なることを示した (5.5 節および A.2 節)。何でもデータが取れればよいのではなく、データ取得の自由度があるとよい。

謝辞 本研究は JSPS 科研費 20H01728 の助成を受けたものである。本稿に考察にあたっては、議論していただいた九州大学の峯恒憲准教授、青山学院大学の松澤芳昭准教授、公立はこだて未来大学の新美礼彦准教授に感謝します。

参考文献

[1] 森永笑子, 松本慎平, 林雄介, 平嶋宗, カード操作方式による学習支援システムにおけるフィードバック方式の提案, 教育システム情報学会 (JSiSE), 第 44 回全国大会, 2019
[2] Amruth N. Kumar, Representing and Evaluating Strategies for Solving Parsons Puzzles, International Conference on Intelligent Tutoring Systems (ITS2019), 2019
[3] 中村 陽太, 大場 みち子, 山口 琢, 伊藤 恵, 学習進度に対応

ⓒ 2020 Information Processing Society of Japan

するパズルを利用したプログラミング思考過程の分析, 情報処理学会研究報告コンピュータと教育 (CE), 2019-09-28
http://id.nii.ac.jp/1001/00199559/
[4] 中村 陽太, 大場 みち子, 山口 琢, 伊藤 恵, 授業進度に対応するパズルを利用したプログラミング思考過程の分析と教育支援システムの開発, 情報処理学会 第 82 回全国大会, 2020
http://id.nii.ac.jp/1001/00205909/
[5] 山口琢, 大場みち子, 編集操作の時間的共起分析の提案, 情報処理学会 研究報告コンピュータと教育 (CE), 2019-CE-151, 2019
http://id.nii.ac.jp/1001/00199567/
[6] 奥村晴彦, ［改訂新版］C 言語による標準アルゴリズム事典, 技術評論社, 2018
https://gihyo.jp/dp/ebook/2018/978-4-7741-9787-6
[7] Yujia Li, Felix Gimeno, Pushmeet Kohli, Oriol Vinyals, Strong Generalization and Efficiency in Neural Programs, arXiv preprint arXiv:2007.03629, 2020
[8] Bengio, Y. & Louradour, Jérôme & Collobert, Ronan & Weston, Jason. Curriculum learning, Journal of the American Podiatry Association. 2009

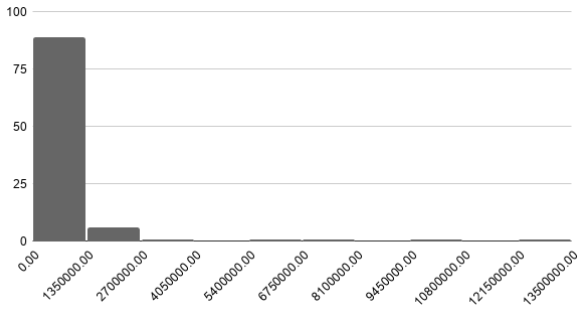

図 **A·1** バブル・ソートの共起頻度の分布

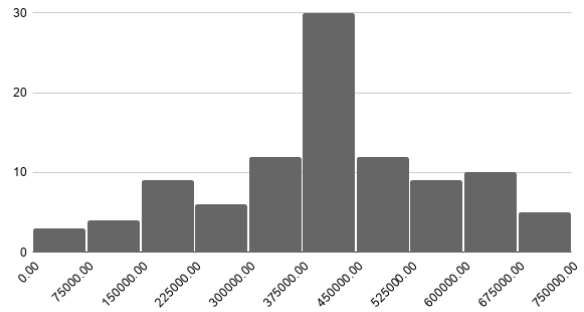

図 **A·2** クイックソートの共起頻度の分布

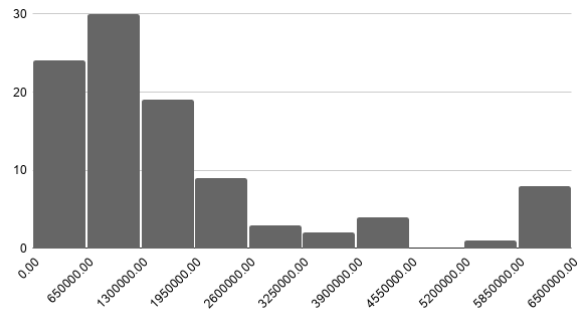

図 **A·3** ヒープソートの共起頻度の分布

付　　録

A.1 コンピューターによる整列操作の共起頻度の分布

図 3 から図 5 までの各共起行列の共起頻度を、それぞれ図 **A·1** から図 **A·3** に示す。セルの数は 10 x 10 = 100 個、すなわち棒の高さの合計は 100 である。

バブル・ソートは、共起頻度の少ないセルの数が多く、グラフは左が高くて右に行くにつれて急に低くなる。突出して頻度の高いセルがある。すなわち、共起頻度の高いセルとそれ以外との頻度の差が大きく、頻度の高いセルの数が極端に少ない (図 A·1)。

クイックソートは、中間程度の頻度のセルが最も多く、少ない方と多い方に裾を広げている (図 **A·2**)。

ヒープソートは、グラフの左側が高く右に行くにつれて

n \ n+1	1	2	3	4	5	6	7	8	9	10
1	0	0	0	0	0	0	0	0	0	0
2	0	0	181,440	192,032	206,240	221,120	235,328	248,288	259,808	269,888
3	0	181,440	256	393,632	422,832	458,378	494,570	528,530	559,010	585,650
4	0	201,600	393,632	4,758	647,112	707,486	774,346	839,218	898,066	949,426
5	0	210,840	436,472	647,112	32,314	959,006	1,066,358	1,174,790	1,274,306	1,360,994
6	0	216,240	459,512	717,192	959,006	127,722	1,351,478	1,520,122	1,676,918	1,813,154
7	0	219,936	474,152	760,392	1,062,686	1,351,478	358,378	1,847,682	2,085,148	2,290,748
8	0	222,740	484,696	790,592	1,134,886	1,497,598	1,847,682	794,178	2,467,727	2,771,705
9	0	225,023	493,031	814,011	1,190,705	1,611,377	2,048,821	2,467,727	1,483,174	3,227,833
10	0	226,980	500,107	833,819	1,238,113	1,708,825	2,222,709	2,742,415	3,227,833	2,436,672

図 A·4 バブル・ソートの共起行列、添字の小さい方を先に記録
平均 780,192, 標準偏差: 814,190.5

n \ n+1	1	2	3	4	5	6	7	8	9	10
1	0	0	0	0	0	0	0	0	0	0
2	0	0	14,400	120,978	117,136	130,516	133,726	146,840	191,419	254,865
3	0	362,640	184,320	186,648	130,102	141,176	158,628	171,908	230,163	304,848
4	0	172,200	526,104	208,080	174,184	163,122	167,412	188,876	261,666	344,027
5	0	143,820	266,088	550,626	216,540	211,272	191,302	205,256	272,639	356,728
6	0	130,008	243,580	400,366	761,404	212,898	251,452	244,154	299,838	419,838
7	0	89,388	184,668	224,986	379,966	810,822	209,372	348,479	369,642	537,785
8	0	69,786	147,710	161,600	226,377	507,206	908,310	237,579	456,309	573,664
9	0	60,708	129,192	139,202	164,963	371,572	612,934	1,078,949	313,850	669,799
10	0	56,370	118,818	129,594	151,088	328,056	507,448	716,825	1,277,686	559,825

図 A·5 クイックソートの共起行列、添字の小さい方を先に記録
平均 243,943.2, 標準偏差: 240,570.2

n \ n+1	1	2	3	4	5	6	7	8	9	10
1	6,869,484	2,634,908	2,857,356	1,976,464	1,900,468	1,455,676	1,953,064	809,576	771,120	334,368
2	190,272	5,381,568	1,566,252	1,931,136	1,592,028	1,484,396	1,354,072	1,116,584	866,160	446,768
3	2,505,296	1,629,616	3,220,104	916,064	1,068,260	1,185,944	912,736	1,133,960	846,720	452,340
4	2,742,444	1,447,452	268,848	1,873,488	619,524	809,664	620,520	995,592	735,264	373,596
5	1,861,152	1,804,372	801,868	236,960	1,192,608	541,440	450,648	771,168	588,816	300,056
6	1,798,084	1,452,916	920,096	475,552	173,216	680,000	358,800	538,560	480,240	325,416
7	1,398,212	1,361,764	1,020,468	619,716	342,200	103,320	310,080	390,240	464,400	387,520
8	1,982,696	1,310,272	802,904	452,528	233,712	113,160	59,520	114,240	609,840	369,128
9	903,184	1,143,008	1,083,144	862,024	550,152	235,680	29,760	0	0	636,248
10	725,760	806,400	766,080	622,080	420,480	218,880	69,120	0	0	0

図 A·6 ヒープソートの共起行列、添字の小さい方を先に記録
平均 991,190.4, 標準偏差: 1,015,469.3

低くなる傾向にある。突出して頻度の高いセルがあるが、バブル・ソートほど極端ではない (図 A·3)。

A.2 添字の小さい方を先に記録したときの共起行列

3.2 節で、ドラッグ&ドロップによる並べ替え操作と比較するために、整列アルゴリズムにおける交換処理を、添字の大きい方を先に動かすとみなして記録することとした。ここで、添字の小さい方を先に動かすとみなして記録した場合の共起行列を図 **A·4** から図 **A·6** に示す。

バブル・ソート (図 A·4) とヒープソート (図 A·6) には突出して頻度の高いセルがある。それ以外には、5 章のような説明を見いだせなかった。

ⓒ 2020 Information Processing Society of Japan