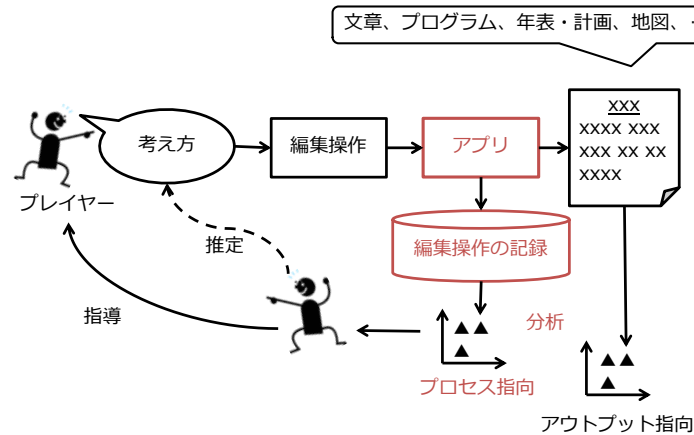


# 人の問題解決プロセスの分析手法を コンピューターに適用する試み

山口 琢(フリーランス) 大場 みち子(京都橘大学)

v1 2025-03-17

# 1. はじめに



人の問題解決プロセス(特に読み書き)を測定・分析する研究

- プロセスをデータ化/測定する手法: デジタル教材など
- 測定データを分析・評価する手法

研究会の趣旨に合ってるかどうか...

## 2. プロセスの研究の進み方

プログラミングのプロセスを評価したい。

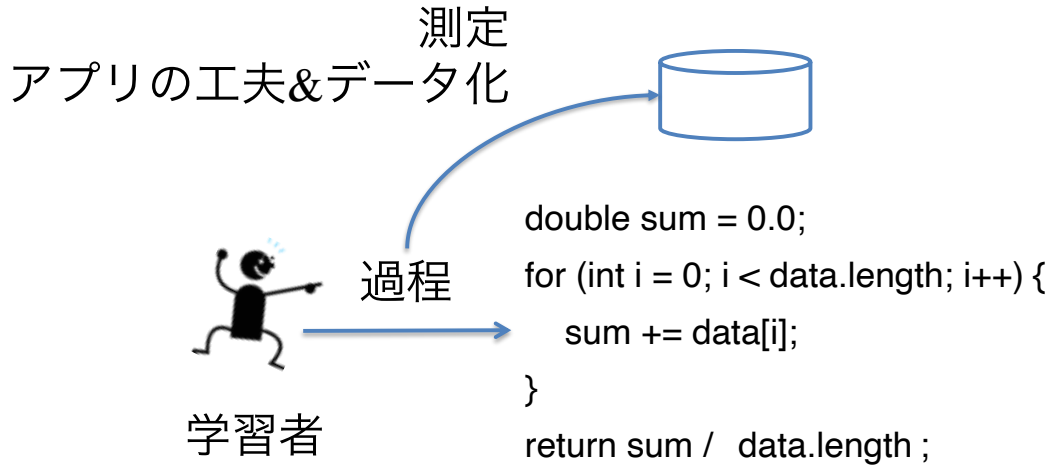


学習者

```
double sum = 0.0;
for (int i = 0; i < data.length; i++) {
    sum += data[i];
}
return sum / data.length;
```

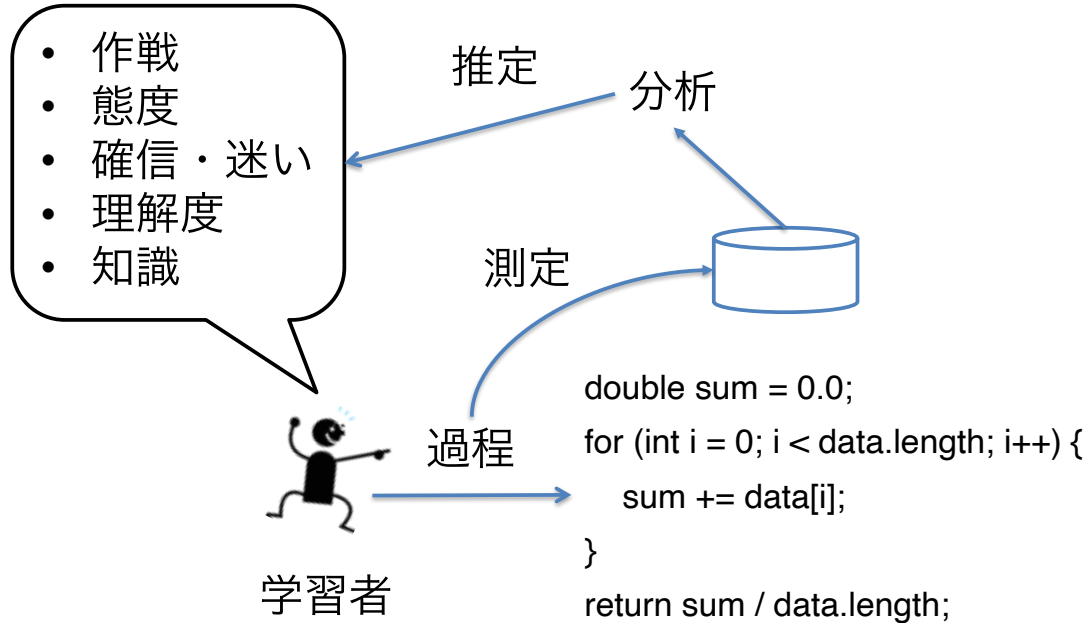
## 2.1 プロセスの研究

プログラミングのアプリを工夫してプロセスをデータ化



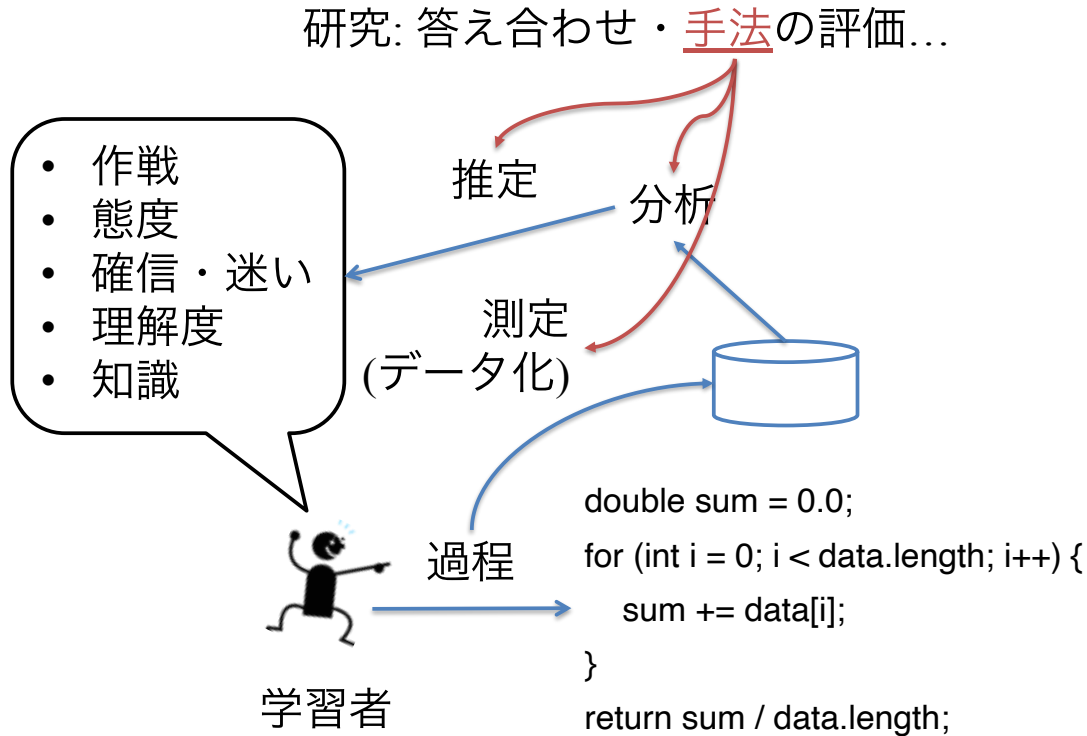
## 2.2 プロセスの研究

測定データ进行分析して推定する。



## 2.3 プロセスの研究

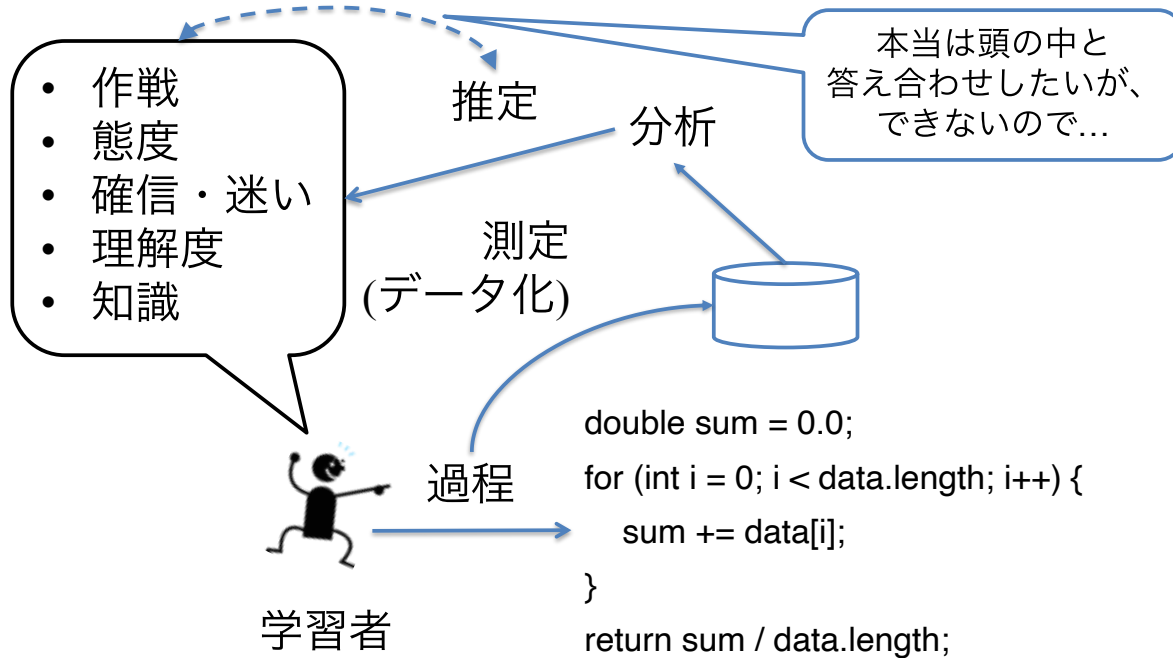
測定、分析、推定を評価



## 2.4 プロセスの研究

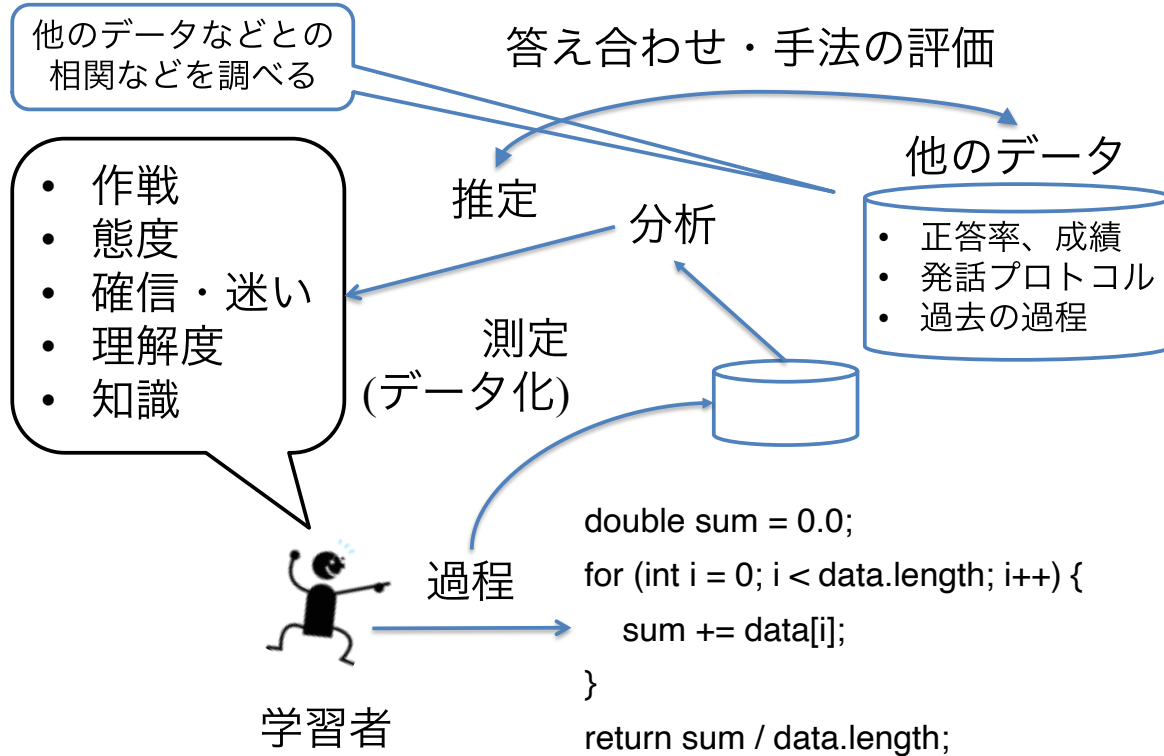
頭の中と答え合わせしたいが、できないので...

答え合わせ・手法の評価...



## 2.5 プロセスの研究

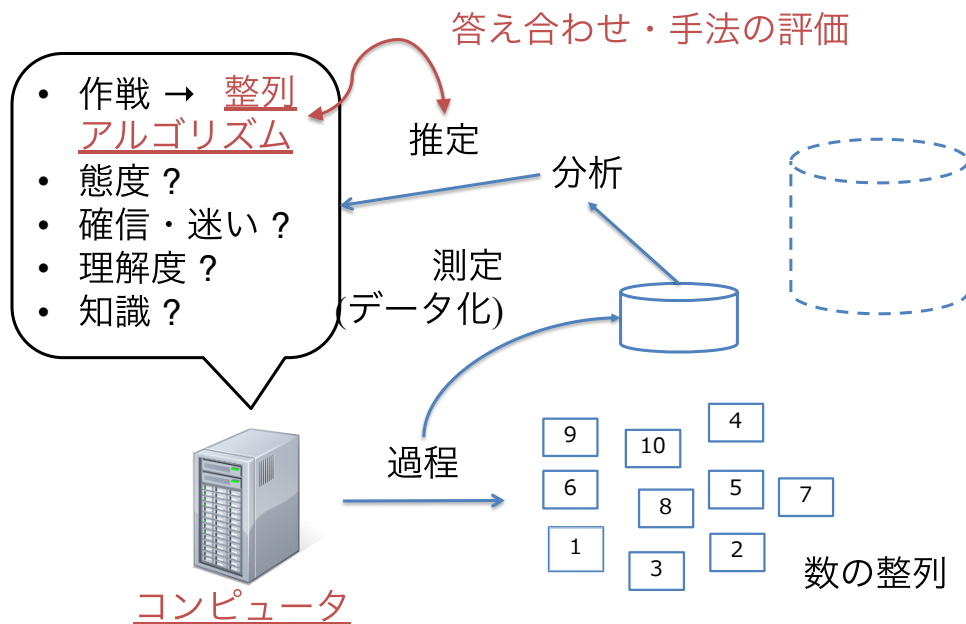
他のデータと答え合わせする。





### 3. 今日のお題

アルゴリズムが分かっているコンピューターによる問題解決プロセスならば、直接答え合わせできるだろう。



## 3.1 アルゴリズムを可視化する教材

ソートの場合

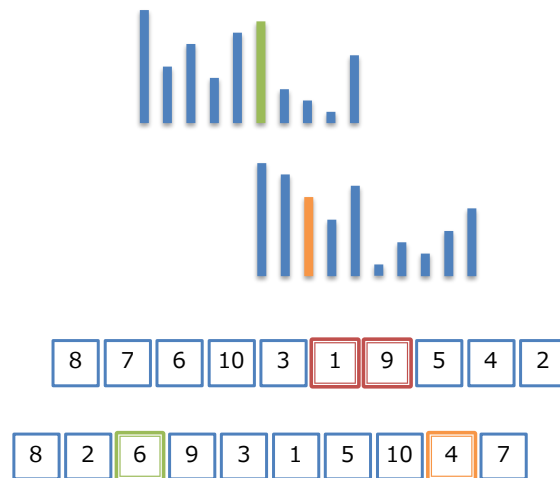
- [バブルソート](#)
- [クイックソート](#)

## 3.2 可視化教材の機能

外から見えるように(可視化)して、処理(アルゴリズム)の理解を助ける

- ソートって何をするの？入力と出力の関係
- 正解してるの？
- **どのように**解いているのか？

可視化

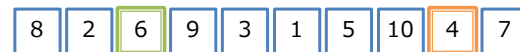


## 3.3 今回のテーマ

逆をやる: 外から見えるモノからプロセスを推定

- 何をしているか？ 入力  
と出力の関係
- 正解してるのか？
- どのように解いている  
のか？

推定



## 4. 方法

問題としてはソートを取り上げて、いくつかの分析手法を試みる。

- 題材: 数のソートを
- 分析手法:
  - 何をしてるか？入力と出力の関係
  - 正解してるのか？
  - 途中の解と**正解 最終的な解答**との距離の変化の分析
  - 操作の時間的な共起分析(temporal co-occurrence)分析
  - 解の状態遷移における閉路(cycle)の分析(今回はスキップ)

## 4.1 方法 > 題材: ソート

入力する問題

- 1 から 10 の 10 個の数
- ${}_{10}P_{10} = 3,628,800$  種類の並びを入力とする

## 4.2 方法 > 題材: ソートのアルゴリズム

- バブルソート
- クイックソート
- ヒープソート
- マージソート
- 選択ソート
- 挿入ソート

## 4.3 方法 > 題材: 測定のタイミング

状態遷移の**状態**はどこか？

アルゴリズム	状態
バブルソート	交換直後の並び
クイックソート	交換直後の並び
ヒープソート	交換直後の並び
選択ソート	交換直後の並び
マージソート	マージ直後の並び
挿入ソート	挿入直後の並び



## 5. 何をしているか？

どのアルゴリズムに、どの問題を与えても、(1,2,3,4,5,6,7,8,9,10) を出力する。

- 推定 → 与えられた10個の数を昇順にソートしてる
- ここで、実際はヒープソートしていても、バブルソートでモデリングしてOK

## 6. 正解なのか？

与えられた10個の数を昇順にソートしてる...とするならば

- いつも正解してる、みんなが正答率100%
- ここで、実際はヒープソートしていても、バブルソートでモデリングしてOK

## 7. 最終的な解答との距離の変化の分析

解答との距離の変化の分析は、

- 問題解決の過程で**変化する解**を記録し、
- ~~正解~~ **最終的な解答**との間に**何らかの距離**を適用して、
- 解くプロセスの時間経過に伴う距離の変化を分析し、
- 解答者の解き方などを推定する

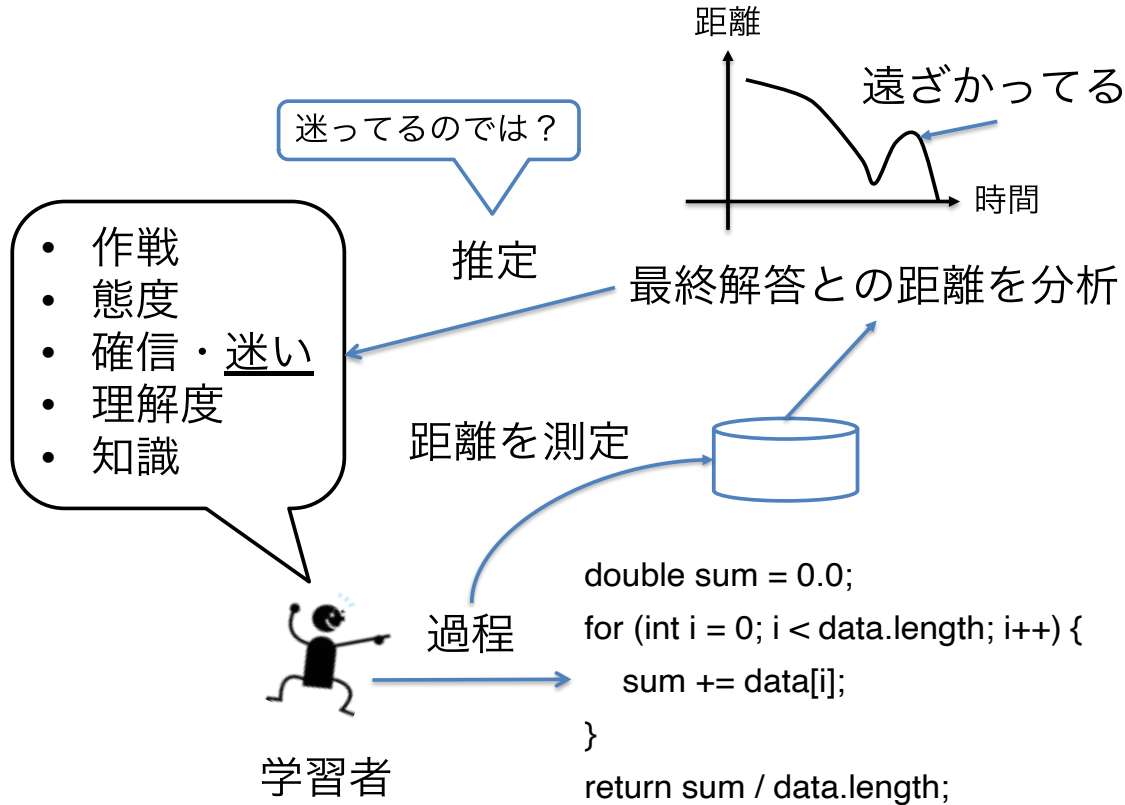
例えば、

**推定:**

途中で最終解答との**距離が大きくなる**ときには、

**解答者が迷っている**と推定する

## 7.1 最終解答との距離の変化の分析



## 7.2 距離 1/2

- ハミング距離(Hamming distance)
- レーベンシュタイン距離(Levenshtein distance)
- ケイリー距離(Cayley distance)
- ウラム距離(Ulam distance)

## 7.3 距離 2/2

- ハミング距離(Hamming distance)

2つの並びの同じ位置にあって異なる要素の個数である。

$(2, 5, 3, 1, 4)$  と  $(1, 2, 3, 4, 5)$  の距離は、3だけが同じで他は異なるので、4である。

- レーベンシュタイン距離(Levenshtein distance)

1要素の挿入・削除・置換(その要素を任意の別の要素に置き換える)操作を何回繰り返すと、一方の並びから他方の並びへ変わるかの操作回数である。断りなく編集距離というとレーベンシュタイン距離を指すことが多い。

- ケイリー距離(Cayley distance)

「交換」操作によってのみ並びを変換して、一方の並びから他方の並びにいたるまでの変換の最小回数を距離とする。

- ウラム距離(Ulam distance)

スマホのUIなどのドラッグ&ドロップに該当する操作をして、目的の並びになるまでの最小の操作数

## 7.4 結果: 状態遷移

swap-vl		levenshtein	hamming	cayley	ulam
start	1, 2, 3, 4, 5, 7, 8,10, 6, 9	4 ####	5 #####	4 ####	2 ##
( 9,10)	1, 2, 3, 4, 5, 7, 8, <u>9</u> , 6, <u>10</u>	2 ##	4 ####	3 ###	1 #
( 6, 8)	1, 2, 3, 4, 5, 7, <u>6</u> , 9, <u>8</u> ,10	3 ###	4 ####	2 ##	2 ##
( 6, 7)	1, 2, 3, 4, 5, <u>6</u> , <u>7</u> , 9, 8,10	2 ##	2 ##	1 #	1 #
( 8, 9)	1, 2, 3, 4, 5, 6, 7, <u>8</u> , <u>9</u> ,10	0	0	0	0
answer	1, 2, 3, 4, 5, 6, 7, 8, 9,10				

- 4 ##...の4が最終解答との距離、##...の長さが目安
- 下線で赤字が操作された数
- 4つの距離について表示

## 7.5 結果: クイックソート

swap-vl		levenshtein	hamming	cayley	ulam
start	1, 2, 3, 4, 5, 7, 8,10, 6, 9	4 ####	5 #####	4 ####	2 ##
( 9,10)	1, 2, 3, 4, 5, 7, 8, <u>9</u> , 6, <u>10</u>	2 ##	4 ####	3 ###	1 #
( 6, 8)	1, 2, 3, 4, 5, 7, <u>6</u> , 9, <u>8</u> ,10	3 ###	4 ####	2 ##	2 ##
( 6, 7)	1, 2, 3, 4, 5, <u>6</u> , <u>7</u> , 9, 8,10	2 ##	2 ##	1 #	1 #
( 8, 9)	1, 2, 3, 4, 5, 6, 7, <u>8</u> , <u>9</u> ,10	0	0	0	0
answer	1, 2, 3, 4, 5, 6, 7, 8, 9,10				

- 途中で遠ざかる場合がある



## 7.6 結果: ヒープソート

```

swap-vl      levenshtein  hamming      cayley      ulam
start    1, 2, 3, 4, 9, 7,10, 5, 6, 8  6 ##### 6 ##### 5 ##### 3 ###
( 2, 1)  2, 1, 3, 4, 9, 7,10, 5, 6, 8  8 ##### 8 ##### 6 ##### 4 ###
( 3, 2)  3, 1, 2, 4, 9, 7,10, 5, 6, 8  8 ##### 9 ##### 7 ##### 4 ###
( 4, 1)  3, 4, 2, 1, 9, 7,10, 5, 6, 8  9 ##### 10 ##### 8 ##### 5 #####
( 4, 3)  4, 3, 2, 1, 9, 7,10, 5, 6, 8  9 ##### 10 ##### 7 ##### 6 #####
( 9, 3)  4, 9, 2, 1, 3, 7,10, 5, 6, 8  9 ##### 10 ##### 8 ##### 5 #####
( 9, 4)  9, 4, 2, 1, 3, 7,10, 5, 6, 8  8 ##### 10 ##### 9 ##### 5 #####
( 7, 2)  9, 4, 7, 1, 3, 2,10, 5, 6, 8  9 ##### 10 ##### 8 ##### 5 #####
(10, 7)  9, 4,10, 1, 3, 2, 7, 5, 6, 8  9 ##### 9 ##### 7 ##### 5 #####
(10, 9) 10, 4, 9, 1, 3, 2, 7, 5, 6, 8  9 ##### 9 ##### 8 ##### 5 #####
( 5, 1) 10, 4, 9, 5, 3, 2, 7, 1, 6, 8  9 ##### 9 ##### 7 ##### 6 #####
( 5, 4) 10, 5, 9, 4, 3, 2, 7, 1, 6, 8  8 ##### 8 ##### 6 ##### 7 #####
( 6, 4) 10, 5, 9, 6, 3, 2, 7, 1, 4, 8  9 ##### 9 ##### 7 ##### 6 #####
( 6, 5) 10, 6, 9, 5, 3, 2, 7, 1, 4, 8  9 ##### 9 ##### 6 ##### 7 #####
( 8, 3) 10, 6, 9, 5, 8, 2, 7, 1, 4, 3  9 ##### 9 ##### 7 ##### 8 #####
( 8, 6) 10, 8, 9, 5, 6, 2, 7, 1, 4, 3  8 ##### 9 ##### 8 ##### 7 #####
(10, 3)  3, 8, 9, 5, 6, 2, 7, 1, 4,10  7 ##### 8 ##### 7 ##### 5 #####
( 9, 3)  9, 8, 3, 5, 6, 2, 7, 1, 4,10  7 ##### 7 ##### 6 ##### 5 #####
( 7, 3)  9, 8, 7, 5, 6, 2, 3, 1, 4,10  8 ##### 9 ##### 7 ##### 6 #####
( 9, 4)  4, 8, 7, 5, 6, 2, 3, 1, 9,10  7 ##### 8 ##### 6 ##### 5 #####
( 8, 4)  8, 4, 7, 5, 6, 2, 3, 1, 9,10  7 ##### 8 ##### 5 ##### 5 #####
( 6, 4)  8, 6, 7, 5, 4, 2, 3, 1, 9,10  8 ##### 8 ##### 4 ##### 6 #####
( 8, 1)  1, 6, 7, 5, 4, 2, 3, 8, 9,10  6 ##### 6 ##### 3 ##### 4 #####
( 7, 1)  7, 6, 1, 5, 4, 2, 3, 8, 9,10  7 ##### 7 ##### 4 ##### 4 #####
( 3, 1)  7, 6, 3, 5, 4, 2, 1, 8, 9,10  6 ##### 6 ##### 3 ##### 5 #####
( 7, 1)  1, 6, 3, 5, 4, 2, 7, 8, 9,10  4 ##### 4 ##### 2 ## 3 ##
( 6, 1)  6, 1, 3, 5, 4, 2, 7, 8, 9,10  5 ##### 5 ##### 3 ## 3 ##
( 5, 1)  6, 5, 3, 1, 4, 2, 7, 8, 9,10  5 ##### 5 ##### 4 ##### 4 #####
( 6, 2)  2, 5, 3, 1, 4, 6, 7, 8, 9,10  4 ##### 4 ##### 3 ## 2 ##
( 5, 2)  5, 2, 3, 1, 4, 6, 7, 8, 9,10  3 ##### 3 ## 2 ## 2 ##
( 4, 2)  5, 4, 3, 1, 2, 6, 7, 8, 9,10  4 ##### 4 ##### 3 ## 3 ##
( 5, 2)  2, 4, 3, 1, 5, 6, 7, 8, 9,10  3 ## 3 ## 2 ## 2 ##
( 4, 2)  4, 2, 3, 1, 5, 6, 7, 8, 9,10  2 ## 2 ## 1 # 2 ##
( 4, 1)  1, 2, 3, 4, 5, 6, 7, 8, 9,10  0 0 0 0
( 3, 1)  3, 2, 1, 4, 5, 6, 7, 8, 9,10  2 ## 2 ## 1 # 2 ##
( 3, 1)  1, 2, 3, 4, 5, 6, 7, 8, 9,10  0 0 0 0
( 2, 1)  2, 1, 3, 4, 5, 6, 7, 8, 9,10  2 ## 2 ## 1 # 1 #
( 2, 1)  1, 2, 3, 4, 5, 6, 7, 8, 9,10  0 0 0 0
answer    1, 2, 3, 4, 5, 6, 7, 8, 9,10

```

## 7.7 結果: ヒープソート

```

swap-vl      levenshtein  hamming      cayley      ulam
start
( 2, 1) 1, 2, 3, 4, 9, 7,10, 5, 6, 8 6 ##### 6 ##### 5 ##### 3 ###
( 3, 2) 2, 1, 3, 4, 9, 7,10, 5, 6, 8 8 ##### 8 ##### 6 ##### 4 #####
( 4, 1) 3, 1, 2, 4, 9, 7,10, 5, 6, 8 8 ##### 9 ##### 7 ##### 4 #####
( 4, 1) 3, 4, 2, 1, 9, 7,10, 5, 6, 8 9 ##### 10 ##### 8 ##### 5 #####
( 4, 3) 4, 3, 2, 1, 9, 7,10, 5, 6, 8 9 ##### 10 ##### 7 ##### 6 #####
( 9, 3) 4, 3, 2, 1, 3, 7,10, 5, 6, 8 8 ##### 10 ##### 8 ##### 5 #####
( 9, 4) 9, 4, 2, 1, 3, 7,10, 5, 6, 8 8 ##### 10 ##### 9 ##### 5 #####
( 7, 2) 9, 4, 7, 1, 3, 2,10, 5, 6, 8 9 ##### 10 ##### 8 ##### 5 #####
(10, 7) 9, 4,10, 1, 3, 2, 7, 5, 6, 8 9 ##### 9 ##### 7 ##### 5 #####
(10, 9) 10, 4, 9, 1, 3, 2, 7, 5, 6, 8 9 ##### 9 ##### 8 ##### 5 #####
( 5, 1) 10, 4, 9, 5, 3, 2, 7, 1, 6, 8 9 ##### 9 ##### 7 ##### 6 #####
( 5, 4) 10, 9, 4, 3, 2, 7, 1, 6, 8 8 ##### 8 ##### 6 ##### 7 #####
( 6, 4) 10, 5, 9, 6, 3, 2, 7, 1, 4, 8 9 ##### 9 ##### 7 ##### 6 #####
( 6, 5) 10, 6, 9, 5, 3, 2, 7, 1, 4, 8 9 ##### 9 ##### 6 ##### 7 #####
( 8, 3) 10, 6, 9, 5, 8, 2, 7, 1, 4, 3 9 ##### 9 ##### 7 ##### 8 #####
( 8, 6) 10, 8, 9, 5, 6, 2, 7, 1, 4, 3 8 ##### 9 ##### 8 ##### 7 #####
(10, 3) 3, 8, 9, 5, 6, 2, 7, 1, 4,10 7 ##### 8 ##### 7 ##### 5 #####
( 9, 3) 9, 8, 3, 5, 6, 2, 7, 1, 4,10 6 ##### 7 ##### 6 ##### 5 #####
( 7, 3) 9, 8, 7, 5, 6, 2, 3, 1, 4,10 7 ##### 9 ##### 7 ##### 6 #####
( 9, 4) 4, 8, 7, 5, 6, 2, 3, 1, 9,10 7 ##### 8 ##### 6 ##### 5 #####
( 8, 4) 8, 4, 7, 5, 6, 2, 3, 1, 9,10 7 ##### 8 ##### 5 ##### 5 #####
( 6, 4) 8, 6, 7, 5, 3, 2, 3, 1, 9,10 8 ##### 8 ##### 4 ##### 6 #####
( 8, 1) 1, 6, 7, 5, 4, 2, 3, 8, 9,10 6 ##### 6 ##### 3 ##### 4 #####
( 7, 1) 7, 6, 3, 5, 4, 2, 3, 8, 9,10 7 ##### 7 ##### 4 ##### 4 #####
( 3, 1) 7, 6, 3, 5, 4, 2, 3, 8, 9,10 6 ##### 6 ##### 3 ##### 5 #####
( 7, 1) 1, 6, 3, 5, 4, 2, 7, 8, 9,10 4 ##### 4 ##### 2 ##### 3 #####
( 6, 1) 6, 3, 5, 4, 2, 7, 8, 9,10 5 ##### 5 ##### 3 ##### 3 #####
( 5, 1) 6, 3, 3, 1, 4, 2, 7, 8, 9,10 5 ##### 5 ##### 4 ##### 4 #####
( 6, 2) 2, 5, 3, 1, 4, 6, 7, 8, 9,10 4 ##### 4 ##### 3 ##### 2 #####
( 5, 2) 5, 2, 3, 1, 4, 6, 7, 8, 9,10 3 ##### 3 ##### 2 ##### 2 #####
( 4, 2) 5, 4, 3, 1, 2, 6, 7, 8, 9,10 4 ##### 4 ##### 3 ##### 3 #####
( 5, 2) 2, 4, 3, 1, 5, 6, 7, 8, 9,10 3 ##### 3 ##### 2 ##### 2 #####
( 4, 2) 4, 2, 3, 1, 5, 6, 7, 8, 9,10 2 ## 2 ## 1 ## 2 ##
( 4, 1) 1, 2, 3, 4, 5, 6, 7, 8, 9,10 0 0 0 0
( 3, 1) 3, 2, 1, 4, 5, 6, 7, 8, 9,10 2 ## 2 ## 1 ## 2 ##
( 3, 1) 1, 2, 3, 4, 5, 6, 7, 8, 9,10 0 0 0 0
( 2, 1) 2, 1, 3, 4, 5, 6, 7, 8, 9,10 2 ## 2 ## 1 ## 1 ##
( 2, 1) 1, 2, 3, 4, 5, 6, 7, 8, 9,10 0 0 0 0
answer 1, 2, 3, 4, 5, 6, 7, 8, 9,10

```

- 途中で遠ざかる場合がある
- 距離=0、すなわち最終解答に達した後でも、遠ざかる場合がある

# 7.8 結果

	Levenshtein 距離		Hamming 距離		Caley 距離		Ulam 距離	
	増加した問題数	割合	増加した問題数	割合	増加した問題数	割合	増加した問題数	割合
バブルソート	2,743,264	75.6%	3,512,825	96.8%	3,624,619	99.9%	0	0.0%
ヒープソート	3,628,800	100.0%	3,628,800	100.0%	3,628,800	100.0%	3,628,800	100.0%
挿入ソート	1,339,176	36.9%	1,964,215	54.1%	3,443,878	94.9%	0	0.0%
マージソート	1,852,755	51.1%	2,432,789	67.0%	3,426,453	94.4%	0	0.0%
クイックソート	1,034,478	28.5%	0	0.0%	1,089,872	30.0%	872,908	24.1%
選択ソート	0	0.0%	0	0.0%	0	0.0%	0	0.0%

解くプロセスの間に、最終解答との距離が増加した問題の数と割合。

## 7.9 解答との距離の変化 > 考察

- ソートのプロセスでは必ずしも距離が減少(単調非増加)しない:
  - 距離が広がることもあるし、
  - 正解に達したにもかかわらず再び誤った順序に変わることもある
- ソートアルゴリズムに**迷い**があるとは思えない
- 解答との距離が遠ざかるからといって「迷ってる」とは、一般的には言えない

しかし、

- 遠ざからない アルゴリズム x 距離 の組み合わせもある
- 距離から何らかの情報を引き出せるのではないか？

## 8. 操作の時間的な共起

操作の時間的な共起分析は、



- 問題解決の過程で行われる操作について、
- いくつかの操作が**時間的に近くで行われる頻度**を集計する。

このように集計する動機は、テキスト分析における語の共起と同様

ある複数の操作が頻繁に時間的に近くで行われるとき、それらの操作は

- 何らかの定型操作であったり、
- 考え方の類似性の手がかりになると解釈できる。

## 8.1 時間的共起の例: 文章の並べ替え作文



### オレオレ詐欺

ひとつは、まさにオレオレ詐欺で、孫や甥などをかたって、金銭を要求する。

このような被害を防ぐ最良の方法は、電話での金銭の要求や、銀行口座やクレジットカードに係わる電話があった場合は、まず、詐欺を疑い、家族や親しい人に相談することだ。

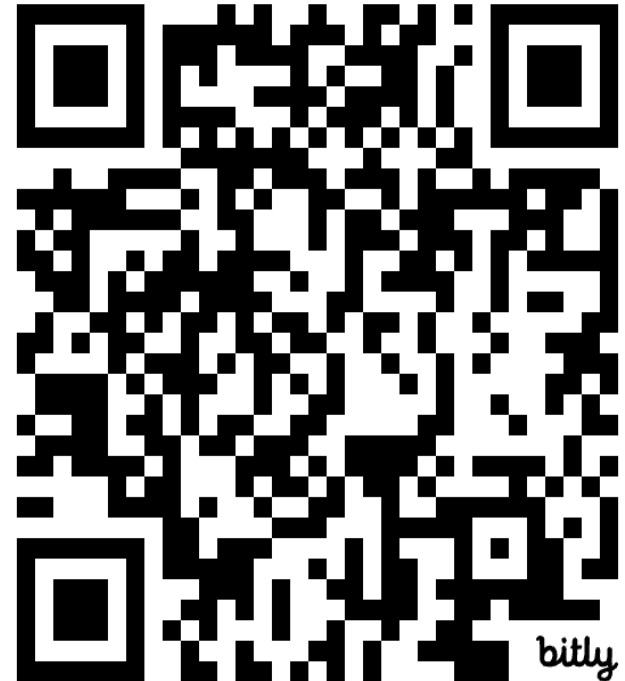
オレオレ詐欺には、大きく分けて2つの種類がある。

ひとつは、官公庁や銀行を騙って、還付金があるとだまして現金自動預払機を操作させ、金銭をだまし取る。

しかし、問題は、身近に相談できる人がいない高齢者が多くいることにあるのかもしれない。

この派生形として、口座が不正に操作されたのだまして、暗証番号を聞き出した上で、銀行カードもだまし取る。

相変わらず、オレオレ詐欺の被害が減らない。



# 8.2 時間的な共起の例: 文章の並べ替え

## 問題と時間的な共起行列

- s2 相変わらず、オレオレ詐欺の被害が減らない。
- s3 オレオレ詐欺には、大きく分けて**2つの種類**がある。
- s4 **ひとつは**、まさにオレオレ詐欺で、孫や甥などをかたって、金銭を要求する。
- s5 **ひとつは**、官公庁や銀行を騙って、還付金があるとだまして現金自動預払機を操作させ、金銭をだまし取る。
- s6 この派生形として、口座が不正に操作されたとだまして、暗証番号を聞き出した上で、銀行カードもだまし取る。
- s7 このような被害を防ぐ最良の方法は、電話での金銭の要求や、銀行口座やクレジットカードに係わる電話があった場合は、まず、詐欺を疑い、家族や親しい人に相談することだ。
- s8 しかし、問題は、身近に相談できる人がいない高齢者が多くいることにあるのかもしれない。

n \ n+1	s2	s3	s4	s5	s6	s7	s8
s2	16	24	6	3	2	6	8
s3	9	16	40	7	0	1	4
s4	5	2	21	30	9	13	3
s5	7	2	6	18	39	8	2
s6	7	1	4	12	14	26	11
s7	5	1	0	7	9	11	38
s8	14	4	4	5	13	5	12

## 8.3 結果: ソート

ソートに時間的共起分析を適用



## 8.4 アルゴリズム B

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

- 同じ数を続けて動かすことが多い
- 10は動かさない

## 8.5 アルゴリズム Q

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

- 「4の次に1」が多い

## 8.6 アルゴリズムH

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

- 次に1つ小さな数を動かすことが多い

## 8.7 アルゴリズム M

n \ n+1	1	2	3	4	5	6	7	8	9	10
1	952,560	1,337,490	990,045	724,455	525,450	377,820	252,090	176,370	92,505	0
2	544,530	771,120	1,289,610	939,960	670,705	466,536	297,978	178,730	93,925	0
3	556,815	431,130	627,480	1,239,750	886,175	611,404	385,965	209,150	94,975	0
4	562,665	433,560	334,470	502,200	1,185,585	826,260	530,001	281,718	109,899	0
5	556,125	429,930	328,600	251,985	383,400	1,123,136	742,560	409,456	152,408	0
6	529,395	412,548	315,422	237,779	179,456	264,960	1,034,640	604,320	235,680	0
7	472,830	372,246	286,254	214,534	156,896	110,160	155,520	877,200	372,360	0
8	374,940	298,380	231,348	173,460	124,512	80,160	53,520	60,480	574,560	0
9	222,390	178,830	139,890	105,234	74,736	43,440	26,760	10,080	0	0
10	0	0	0	0	0	0	0	0	0	0

- 次に1つ大きな数を動かすことが多い

## 8.8 アルゴリズム S

$n \setminus n+1$	1	2	3	4	5	6	7	8	9	10
1	0	1,048,320	322,560	322,560	322,560	322,560	322,560	322,560	322,560	322,560
2	0	322,560	1,159,200	352,800	352,800	352,800	352,800	352,800	352,800	352,800
3	0	322,560	352,800	1,296,000	388,800	388,800	388,800	388,800	388,800	388,800
4	0	322,560	352,800	388,800	1,468,800	432,000	432,000	432,000	432,000	432,000
5	0	322,560	352,800	388,800	432,000	1,693,440	483,840	483,840	483,840	483,840
6	0	322,560	352,800	388,800	432,000	483,840	1,995,840	544,320	544,320	544,320
7	0	322,560	352,800	388,800	432,000	483,840	544,320	2,419,200	604,800	604,800
8	0	322,560	352,800	388,800	432,000	483,840	544,320	604,800	3,024,000	604,800
9	0	322,560	352,800	388,800	432,000	483,840	544,320	604,800	604,800	0
10	0	322,560	352,800	388,800	432,000	483,840	544,320	604,800	604,800	0

- 次に1つ大きな数を動かすことが多い
- 次に1が動かされることがない

## 8.9 アルゴリズムI

n \ n+1	1	2	3	4	5	6	7	8	9	10
1	0	410,553	405,513	398,313	387,873	372,321	348,075	307,179	226,980	0
2	410,553	0	396,853	390,373	380,773	366,205	343,117	303,591	225,023	0
3	396,853	396,853	0	381,196	372,556	359,116	337,360	299,416	222,740	0
4	381,196	381,196	381,196	0	362,604	350,508	330,348	294,312	219,936	0
5	362,604	362,604	362,604	362,604	0	339,384	321,240	287,640	216,240	0
6	339,384	339,384	339,384	339,384	339,384	0	308,280	278,040	210,840	0
7	308,280	308,280	308,280	308,280	308,280	308,280	0	262,080	201,600	0
8	262,080	262,080	262,080	262,080	262,080	262,080	262,080	0	181,440	0
9	181,440	181,440	181,440	181,440	181,440	181,440	181,440	181,440	0	0
10	0	0	0	0	0	0	0	0	0	0

- 同じ数を続けて動かさない
- 10を動かさない

# 8.10 答え合わせ: アルゴリズム B

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

## バブルソート

- 同じ数を続けて動かすことが多い
- 10は動かさない

## 8.11 答え合わせ: アルゴリズム Q

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

クイックソート

- 「4の次に1」が多い



## 8.12 答え合わせ: アルゴリズム H

$n \backslash n+1$	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

### ヒープソート

- 次に1つ小さな数を動かすことが多い

## 8.13 答え合わせ: アルゴリズム M

n \ n+1	1	2	3	4	5	6	7	8	9	10
1	952,560	1,337,490	990,045	724,455	525,450	377,820	252,090	176,370	92,505	0
2	544,530	771,120	1,289,610	939,960	670,705	466,536	297,978	178,730	93,925	0
3	556,815	431,130	627,480	1,239,750	886,175	611,404	385,965	209,150	94,975	0
4	562,665	433,560	334,470	502,200	1,185,585	826,260	530,001	281,718	109,899	0
5	556,125	429,930	328,600	251,985	383,400	1,123,136	742,560	409,456	152,408	0
6	529,395	412,548	315,422	237,779	179,456	264,960	1,034,640	604,320	235,680	0
7	472,830	372,246	286,254	214,534	156,896	110,160	155,520	877,200	372,360	0
8	374,940	298,380	231,348	173,460	124,512	80,160	53,520	60,480	574,560	0
9	222,390	178,830	139,890	105,234	74,736	43,440	26,760	10,080	0	0
10	0	0	0	0	0	0	0	0	0	0

マージソート

- 次に1つ大きな数を動かすことが多い

# 8.14 答え合わせ: アルゴリズム S

n \ n+1	1	2	3	4	5	6	7	8	9	10
1	0	1,048,320	322,560	322,560	322,560	322,560	322,560	322,560	322,560	322,560
2	0	322,560	1,159,200	352,800	352,800	352,800	352,800	352,800	352,800	352,800
3	0	322,560	352,800	1,296,000	388,800	388,800	388,800	388,800	388,800	388,800
4	0	322,560	352,800	388,800	1,468,800	432,000	432,000	432,000	432,000	432,000
5	0	322,560	352,800	388,800	432,000	1,693,440	483,840	483,840	483,840	483,840
6	0	322,560	352,800	388,800	432,000	483,840	1,995,840	544,320	544,320	544,320
7	0	322,560	352,800	388,800	432,000	483,840	544,320	2,419,200	604,800	604,800
8	0	322,560	352,800	388,800	432,000	483,840	544,320	604,800	3,024,000	604,800
9	0	322,560	352,800	388,800	432,000	483,840	544,320	604,800	604,800	0
10	0	322,560	352,800	388,800	432,000	483,840	544,320	604,800	604,800	0

## 選択ソート

- 次に1つ大きな数を動かすことが多い
- 次に1が動かされることがない

# 8.15 答え合わせ: アルゴリズムI

n \ n+1	1	2	3	4	5	6	7	8	9	10
1	0	410,553	405,513	398,313	387,873	372,321	348,075	307,179	226,980	0
2	410,553	0	396,853	390,373	380,773	366,205	343,117	303,591	225,023	0
3	396,853	396,853	0	381,196	372,556	359,116	337,360	299,416	222,740	0
4	381,196	381,196	381,196	0	362,604	350,508	330,348	294,312	219,936	0
5	362,604	362,604	362,604	362,604	0	339,384	321,240	287,640	216,240	0
6	339,384	339,384	339,384	339,384	339,384	0	308,280	278,040	210,840	0
7	308,280	308,280	308,280	308,280	308,280	308,280	0	262,080	201,600	0
8	262,080	262,080	262,080	262,080	262,080	262,080	262,080	0	181,440	0
9	181,440	181,440	181,440	181,440	181,440	181,440	181,440	181,440	0	0
10	0	0	0	0	0	0	0	0	0	0

## 挿入ソート

- 同じ数を続けて動かさない

## 8.16 操作の時間的な共起 > 考察

時間的な共起分析は:

- ソート・アルゴリズムの特徴を表している:
- アルゴリズムそのものまでは遠い...

コンピューター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って表現できた。

## 9. 全体の考察: 正解の存在

### 正解の観点から手法の特徴

- 唯一の正解が必要
- 正解が複数存在してもよい
- 正解が誰にも分からなくてもよい
- 正解が存在しなくてもよい

### 正解を前提としない分析

- 解答との距離の変化の分析
- 操作の時間的な共起
- 状態遷移の閉路(今回はスキップ)

## 10. 全体の考察: このような研究の意義

- 学校教育
  - 「答えのない課題解決に挑む」人材の育成
  - 試行錯誤の検出、内容の評価
- オフィス・アプリケーション
  - 情報の一貫性(coherence)、整合性(consistency)をもたらしそうとする**操作**に基づく品質評価  
vs. アウトプットの品質

# 11. 面白そう

- 他の問題: 人にも解けて、(既知のアルゴリズムで)コンピューターにも解ける

- 15パズル、解けない問題もある

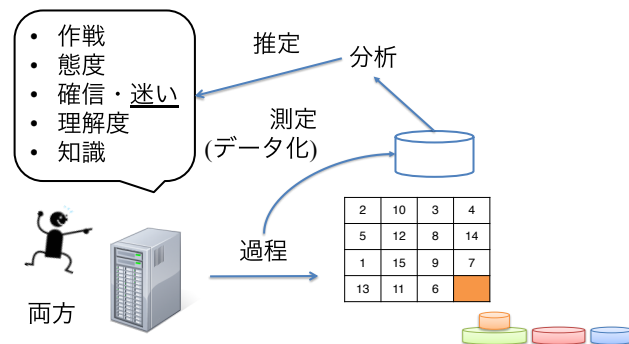
- ハノイの塔

- 上達すると、**その人にとっての距離**が変わる...

その人にとっての距離の変化 ∈ **理解**

...かも

- 言語モデルに「オレオレ詐欺」を解かせる





以上