

アルゴリズム論 (第3回)



岩手県立大学
Iwate Prefectural University

佐々木研(情報システム構築学講座)

講師 山田敬三

k-yamada@iwate-pu.ac.jp

単純挿入法



岩手県立大学
Iwate Prefectural University

単純挿入法

- 注目している部分列の先頭要素を,
既に整列された部分列の適切な場所に挿入する.

単純挿入法のアルゴリズム

for $i \leftarrow 2$ to n do

if a_0, \dots, a_{i-1} の中に, $a_i < a_j$ となる a_j が存在する then

条件を満たす最初の a_j の直前に a_i を挿入する.

end

end

実行例

a[] = 7, 2, 3, 4, 8, 1, 5, 6

a[] = 2, 7, 3, 4, 8, 1, 5, 6

a[] = 2, 3, 7, 4, 8, 1, 5, 6

a[] = 2, 3, 4, 7, 8, 1, 5, 6

a[] = 2, 3, 4, 7, 8, 1, 5, 6

a[] = 1, 2, 3, 4, 7, 8, 5, 6

a[] = 1, 2, 3, 4, 5, 7, 8, 6

a[] = 1, 2, 3, 4, 5, 6, 7, 8

単純選択法の計算量

- 比較回数

$$C \leq 1 + 2 + \dots + (n - 1) = O(n^2)$$

- 挿入回数

$$N \leq n - 1$$

基数ソート



岩手県立大学
Iwate Prefectural University

基数ソート

- キーが整数のときに利用できる.
- キーの最大桁数を k とする. (最大値が1234のとき, $k = 4$)

基数ソートのアルゴリズム

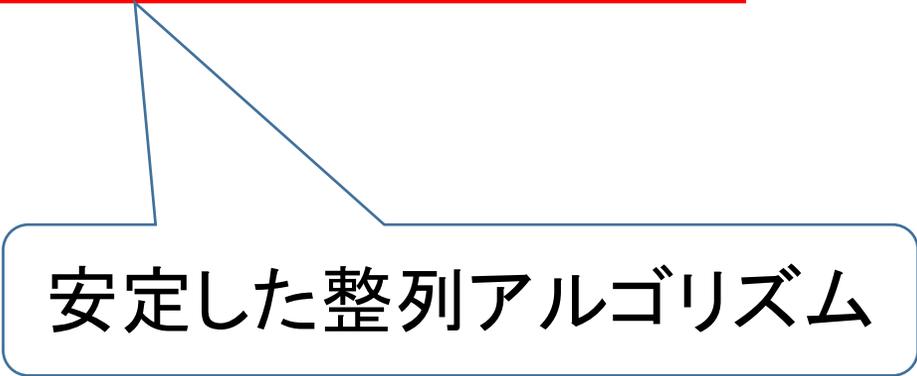
for $j \leftarrow 1$ to k do

列を下位から第 j 桁目に関して整列する.

ただし, 第 j 桁目の値が同じ要素については,

それらの順番が整列の前後で変わらないようにする.

end



安定した整列アルゴリズム

実行例

入力列: 125, 900, 123, 800, 143, 700, 600, 501

一の桁: 900, 800, 700, 600, 501, 123, 143, 125

十の桁: 900, 800, 700, 600, 501, 123, 125, 143

百の桁: 123, 125, 143, 501, 600, 700, 800, 900

バケットソート



岩手県立大学
Iwate Prefectural University

バケットソート

- 入力列を $key(a_i)$ の昇順に整列した列を返す.

バケットソートのアルゴリズム

各 $0 \leq x \leq 9$ について, P_x を用意する.

for $i \leftarrow 1$ to n do

$y \leftarrow \text{key}(a_i)$

a_i を P_y の末尾に追加する.

end

P_0 から P_9 までを接続し, 出力する.

実行例 ($key(a_i)$ は一の桁)

入力列: 125, 900, 123, 800, 143, 700, 600, 501

P_0 : 900, 800, 700, 600

P_1 : 501

P_3 : 123, 143

P_5 : 125

出力列: 900, 800, 700, 600, 501, 123, 143, 125

ソート法まとめ

	比較による	時間計算量	空間計算量	安定性
単純選択ソート	有	$O(N^2)$	$O(1)$	無
単純挿入ソート	有	$O(N^2)$	$O(1)$	有
バブルソート	有	$O(N^2)$	$O(1)$	有
クイックソート	有	$O(N \log N)$	$O(\log N)$	無
基数ソート	無	$O(N)$	$O(N)$	有

問題

- 次の配列を単純選択法, 単純挿入法, バブルソート, クイックソート (最初の分割のみ, pivotは任意)で, それぞれ整列しなさい.

8, 4, 6, 5, 1, 7, 2

- 次の配列を基数ソートを用いて, 整列しなさい.

851, 367, 783, 478, 143, 117, 127, 509