

色々なソート 1

目次

- バブルソート
- 選択ソート
- 挿入ソート
- ヒープソート
- まとめ

バブルソート

- 隣り合う2つの数字を比較して入れ替えていくソート方法
- 計算時間は $O(n^2)$ （比較回数が $n, n-1, n-2, \dots, 1$ と減っていくため）

バブルソートの図

https://www.codereading.com/algorithm_and_ds/algorithm/bubble_sort.html

ソート前の配列



1. 隣り合う要素を比較する。交換しない。



2. 隣り合う要素を右に進めて比較する。交換する。



3. もう一度隣り合う要素を右に進めて比較する。交換する。



4. 1度目のソートが終わった状態。

ソート済みの終端の配列を除外して1から3を繰り返す。



色の意味



選択ソート

- 最小値を線形探索し、未確定の先頭データと入れ替えていく方法
- 計算時間は $\mathcal{O}(n^2)$ （比較回数が $n, n-1, n-2, \dots, 1$ と減っていくため）

選択ソートの図

https://www.codereading.com/algorithm_and_ds/algorithm/selection_sort.html

ソート前の配列



1. 初回はすべてが未整列である。最小値「1」を先頭に移動する。(「3」と交換)



2. 未整列の範囲を右に進めて最小値「2」を探し先頭に移動する。(「9」と交換)



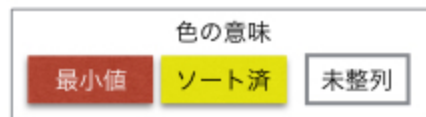
3. 未整列の範囲を右に進め最小値「3」を探し先頭に移動する。(「6」と交換)



4. 未整列の範囲を右に進め最小値「6」を探し先頭に移動する。(交換しない)



ソート完了



挿入ソート

- 未確定のデータ領域のデータを一つ持ってきて、ソート済みのデータに挿入していくソート方法
- 計算時間は $O(n^2)$ （比較回数が $n, n-1, n-2, \dots, 1$ と減っていくため）

挿入ソートの図

https://www.codereading.com/algorithm_and_ds/algorithm/insertion_sort.html

ソート前の配列



1. ソート済み「3」とソートされていない要素「9」を比較する。
 $3 < 9$ なので「9」は「3」の後ろに挿入される。



2. ソート済みの要素「9」とソートされていない要素「6」を比較する。
 $9 > 6$ なので「6」は「3」と「9」の間に挿入される。



3. 「1」は先頭に挿入される。



4. 「2」は「1」と「3」の間に挿入される。



ソート完了



色の意味

ソート済

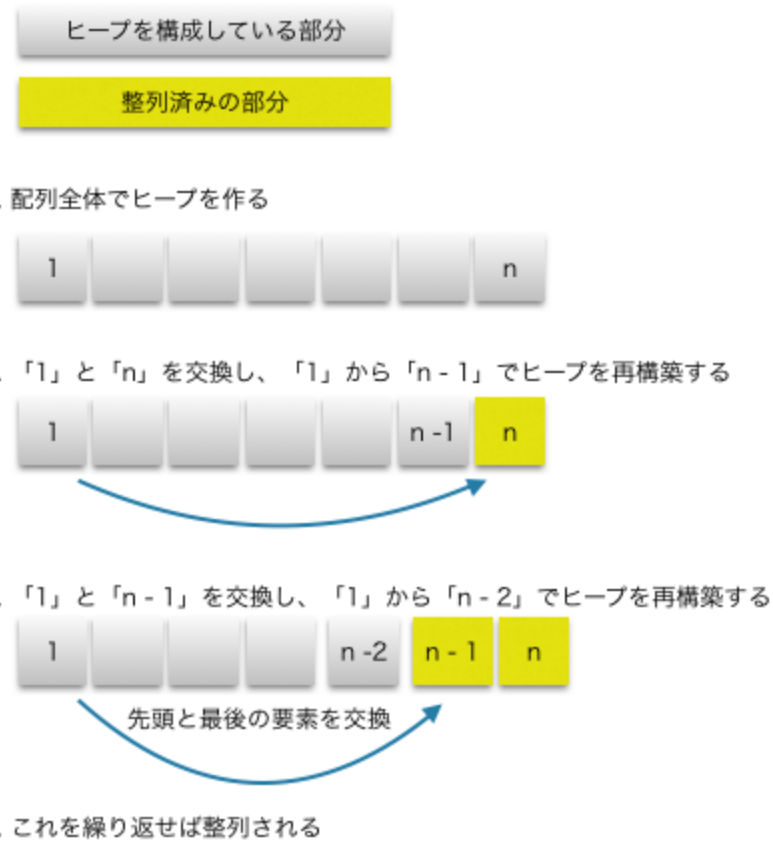
ソートされていない

ヒープソート

- ヒープの先頭データ（最小値か最大値となる）をポップしていくソート方法
- 計算時間は $O(n \log n)$ （ n 個の数字からヒープを作る時間 + ヒープを再構築する時間）

挿入ソートの図

https://www.codereading.com/algorithm_and_ds/algorithm/heap_sort.html



まとめ

- バブルソート、選択ソート、挿入ソート、ヒープソートを説明、実装
- とくに、ヒープソートの実装が難しかった。。。。

参考URL・書籍

ソート全般について: <https://www.codereading.com/>

ヒープソートについて: <https://novnote.com/priority-queue-heapsort-impl-cpp/565/>

「アルゴリズム図鑑」 石田保輝, 宮崎修一著 (翔泳社)

次回につづく (マージンソート、クイックソートなど)