

CS 112 - Fall 2012, Lab 02

Haohan Zhu

Sorting Algorithm

- Selection Sort
 - Insertion Sort
 - Merge Sort
 - Quick Sort
-
- Sorting algorithm from the lectures

Sorting Algorithm

| | Best | Average | Worst | Memory |
|----------------|------------|------------|------------|----------|
| Selection sort | n^2 | n^2 | n^2 | 1 |
| Insertion sort | n | n^2 | n^2 | 1 |
| Merge sort | $n \log n$ | $n \log n$ | $n \log n$ | n |
| Quick sort | $n \log n$ | $n \log n$ | n^2 | $\log n$ |

Sorting Algorithm

- Shell sort
- Heapsort

- Sorting algorithm from the textbook

Sorting Algorithm

- Bubble sort
- Cocktail sort
- Timsort (Java)
- ...

- Other sorting algorithms

Sorting Algorithm

- Sorting algorithms on Wikipedia
- Popular algorithms but not a full list

| Comparison sorts | | | | | | | |
|---------------------|------------|----------------------------|--|----------------------------|---------|--------------------------|--|
| Name | Best | Average | Worst | Memory | Stable | Method | Other notes |
| Binary tree sort | n | $n \log n$ | $n \log n$ | n | Yes | Insertion | When using a self-balancing binary search tree |
| Bogosort | n | $n \cdot n!$ | $n \cdot n! \rightarrow \infty$ | 1 | No | Luck | Randomly permute the array and check if sorted. |
| Bubble sort | n | n^2 | n^2 | 1 | Yes | Exchanging | Tiny code size |
| Cocktail sort | n | n^2 | n^2 | 1 | Yes | Exchanging | |
| Comb sort | n | $n \log n$ | n^2 | 1 | No | Exchanging | Small code size |
| Cycle sort | — | n^2 | n^2 | 1 | No | Insertion | In-place with theoretically optimal number of writes |
| Gnome sort | n | n^2 | n^2 | 1 | Yes | Exchanging | Tiny code size |
| Heapsort | $n \log n$ | $n \log n$ | $n \log n$ | 1 | No | Selection | |
| In-place Merge sort | — | — | $n (\log n)^2$ | 1 | Yes | Merging | Implemented in Standard Template Library (STL); ^[2] can be implemented as a stable sort based on stable in-place merging. ^[5] |
| Insertion sort | n | n^2 | n^2 | 1 | Yes | Insertion | $O(n + d)$, where d is the number of inversions |
| Introsort | $n \log n$ | $n \log n$ | $n \log n$ | $\log n$ | No | Partitioning & Selection | Used in SGI STL implementations |
| Library sort | — | $n \log n$ | n^2 | n | Yes | Insertion | |
| Merge sort | $n \log n$ | $n \log n$ | $n \log n$ | Depends; worst case is n | Yes | Merging | Highly parallelizable (up to $O(\log(n))$) for processing large amounts of data. |
| Patience sorting | — | — | $n \log n$ | n | No | Insertion & Selection | Finds all the longest increasing subsequences within $O(n \log n)$ |
| Quicksort | $n \log n$ | $n \log n$ | n^2 | $\log n$ | Depends | Partitioning | Quicksort is usually done in place with $O(\log(n))$ stack space. ^[citation needed] Most implementations are unstable, as stable in-place partitioning is more complex. Naive variants use an $O(n)$ space array to store the partition. ^[citation needed] |
| Selection sort | n^2 | n^2 | n^2 | 1 | No | Selection | Stable with $O(n)$ extra space, for example using lists. ^[4] Used to sort this table in Safari or other Webkit web browser. ^[5] |
| Shell sort | n | $n(\log n)^2$ or $n^{3/2}$ | Depends on gap sequence, best known is $n(\log n)^2$ | 1 | No | Insertion | |
| Smoothsort | n | $n \log n$ | $n \log n$ | 1 | No | Selection | An adaptive sort - n comparisons when the data is already sorted, and 0 swaps. |
| Strand sort | n | n^2 | n^2 | n | Yes | Selection | |
| Timsort | n | $n \log n$ | $n \log n$ | n | Yes | Insertion & Merging | n comparisons when the data is already sorted or reverse sorted. |
| Tournament sort | — | $n \log n$ | $n \log n$ | | | Selection | |

Sorting Algorithm in Java

- Class Collections
 - `sort(List<T> list)`
 - `sort(List<T> list, Comparator<? super T> c)`

- Class Arrays
 - `sort()` for Primitive Types and Objects

- PriorityQueue

Practice

- Implementing Bubble Sort
- Compare two adjacent elements and sort them

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 7 | 1 | 5 | 3 | 8 | 4 | 6 | 2 |
|---|---|---|---|---|---|---|---|

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 1 | 7 | 5 | 3 | 8 | 4 | 6 | 2 |
|---|---|---|---|---|---|---|---|

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 1 | 5 | 7 | 3 | 8 | 4 | 6 | 2 |
|---|---|---|---|---|---|---|---|

Practice

- Bubble Sort or Cocktail Sort
- Different directions

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 7 | 1 | 5 | 3 | 8 | 4 | 6 | 2 |
|---|---|---|---|---|---|---|---|

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 1 | 7 | 5 | 3 | 8 | 4 | 6 | 2 |
|---|---|---|---|---|---|---|---|

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 1 | 5 | 7 | 3 | 8 | 4 | 6 | 2 |
|---|---|---|---|---|---|---|---|