Sorting Algorithm Animations

Problem Size: 20 · 30 · 40 · 50 Magnification: 1x · 2x · 3x

Algorithm: Insertion · Selection · Bubble · Shell · Merge · Heap · Quick · Quick3

Initial Condition: Random · Nearly Sorted · Reversed · Few Unique

	Insertion	Selection	Bubble	Shell	Merge	Heap	Quick	Quick3
Random								
Nearly Sorted								
Reversed								
Few Unique								

Discussion

These pages show 8 different sorting algorithms on 4 different initial conditions. These visualizations are intended to:

- Show how each algorithm operates.
- Show that there is no best sorting algorithm.
- Show the advantages and disadvantages of each algorithm.
- Show that worse-case asymptotic behavior is not always the deciding factor in choosing an algorithm.
- Show that the initial condition (input order and key distribution) affects performance as much as the algorithm choice.

The ideal sorting algorithm would have the following properties:

- Stable: Equal keys aren't reordered.
- Operates in place, requiring O(1) extra space.
- Worst-case O(n·lg(n)) key comparisons.
- Worst-case O(n) swaps.
- Adaptive: Speeds up to O(n) when data is nearly sorted or when there are few unique keys.

There is no algorithm that has all of these properties, and so the choice of sorting algorithm depends on the application.

Sorting is a vast topic; this site explores the topic of in-memory generic algorithms for arrays. External sorting, radix sorting, string sorting, and linked list sorting—all wonderful and interesting topics—are deliberately omitted to limit the scope of discussion.

Directions

- Click on above to restart the animations in a row, a column, or the entire table.
- Click directly on an animation image to start or restart it.
- Click on a problem size number to reset all animations.

Key

- Black values are sorted.
- Gray values are unsorted.
- A red triangle marks the algorithm position.
- Dark gray values denote the current interval (shell, merge, quick).
- A pair of red triangles marks the left and right pointers (quick).

References

Algorithms in Java, Parts 1-4, 3rd edition by Robert Sedgewick. Addison Wesley, 2003.

Programming Pearls by Jon Bentley. Addison Wesley, 1986.

Quicksort is Optimal by Robert Sedgewick and Jon Bentley, Knuthfest, Stanford University, January, 2002.

Dual Pivot Quicksort: Code and Discussion.

Bubble-sort with Hungarian ("Csángó") folk dance YouTube video, created at Sapientia University, Tirgu Mures (Marosvásárhely), Romania.

Select-sort with Gypsy folk dance YouTube video, created at Sapientia University, Tirgu Mures (Marosvásárhely), Romania.

Sorting Out Sorting, Ronald M. Baecker with the assistance of David Sherman, 30 minute color sound film, Dynamic Graphics Project, University of Toronto, 1981. Excerpted and reprinted in SIGGRAPH Video Review 7, 1983. Distributed by Morgan Kaufmann, Publishers. Excerpt.