<https://en.wikipedia.org/wiki/Radix_sort#An_example>

|  |  |
| --- | --- |
| **Radix sort** | |
| **Class** | [Sorting algorithm](https://en.wikipedia.org/wiki/Sorting_algorithm) |
| **Data structure** | [Array](https://en.wikipedia.org/wiki/Array_data_type) |
| [**Worst-case performance**](https://en.wikipedia.org/wiki/Best,_worst_and_average_case) | *O*(*wn*) |
| [**Worst-case space complexity**](https://en.wikipedia.org/wiki/Best,_worst_and_average_case) | *O*(*w* + *N*) |

In [computer science](https://en.wikipedia.org/wiki/Computer_science), **radix sort** is a non-[comparative](https://en.wikipedia.org/wiki/Comparison_sort) [integer](https://en.wikipedia.org/wiki/Integer_sorting) [sorting algorithm](https://en.wikipedia.org/wiki/Sorting_algorithm) that sorts data with integer keys by grouping keys by the individual digits which share the same [significant](https://en.wikipedia.org/wiki/Significant_figures) position and value. A [positional notation](https://en.wikipedia.org/wiki/Positional_notation) is required, but because integers can represent strings of characters (e.g., names or dates) and specially formatted floating point numbers, [radix](https://en.wikipedia.org/wiki/Radix) sort is not limited to integers. Radix sort dates back as far as 1887 to the work of [Herman Hollerith](https://en.wikipedia.org/wiki/Herman_Hollerith) on [tabulating machines](https://en.wikipedia.org/wiki/Tabulating_machines).[[1]](https://en.wikipedia.org/wiki/Radix_sort#cite_note-1)

Most digital computers internally represent all of their data as electronic representations of binary numbers, so processing the digits of integer representations by groups of binary digit representations is most convenient. Two classifications of radix sorts are [least significant digit](https://en.wikipedia.org/wiki/Least_significant_digit) (LSD) radix sorts and [most significant digit](https://en.wikipedia.org/wiki/Most_significant_digit)(MSD) radix sorts. LSD radix sorts process the integer representations starting from the least significant digit and move towards the most significant digit. MSD radix sorts work the other way around.

LSD radix sorts typically use the following sorting order: short keys come before longer keys, and keys of the same length are sorted lexicographically. This coincides with the normal order of integer representations, such as the sequence 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11.

MSD radix sorts use lexicographic order, which is suitable for sorting strings, such as words, or fixed-length integer representations. A sequence such as "b, c, d, e, f, g, h, i, j, ba" would be lexicographically sorted as "b, ba, c, d, e, f, g, h, i, j". If lexicographic ordering is used to sort variable-length integer representations, then the representations of the numbers from 1 to 10 would be output as 1, 10, 2, 3, 4, 5, 6, 7, 8, 9, as if the shorter keys were left-justified and padded on the right with blank characters to make the shorter keys as long as the longest key for the purpose of determining sorted order.