

Radix Sort

- make assumptions about the data
- data must have same radix and width
- because of this, data must be integers or strings
- Sort based on each individual digit or letter position
- start at the rightmost position
- must use a stable sort algorithm at each stage
- time complexity $O(n)$
- stable algorithm
- $O(n)$
- it often runs slower than $O(n \log n)$ algorithms because of the overhead involved

For example:

Radix :

Radix for decimal system is 10

- Because there are 10 possible digits in the decimal system
- 0 – 9

Radix for binary numbers is 2

- Because we use two digits in the binary system
- 0, 1

Radix for English alphabet is 26

- Because there are 26 letters in the alphabet

Width:

Width for number 1234 is 4

Width for string “hello” is 5

Width for number 10 is 2

We sort by the element that have least significant digit to the most

RADIX SORT MUST BE STABLE TO REACH CORRECT SORTED ORDER

4725	4586	1330	8792	1594	5729	DESC
1330	8792	1594	4725	4586	5729	SORT BY UNITS

4725	5729	1330	4586	8792	1594	SORT BY TENS
1330	4586	1594	4725	5729	8792	SORT BY HUNDREDS
1330	1594	4586	4725	5729	8792	SORT BY THOUSANDS