

Radix Sort

based on sorting on keys

keys are digits of a number (while sorting array)

eg:

Sort

225, 315, 221, 326, 216

Step 1
assign numbers to their group:

	221				215, 315	326, 216			
0	1	2	3	4	5	6	7	8	9

if two numbers have same key value they are inserted in dictionary in the order they came.

flatten

221, 215, 315, 326, 216

221, 215, 315, 326, 216

Step 2

	215, 315, 216	221, 326							
0	1	2	3	4	5	6	7	8	9

flatten :

215, 315, 216, 221, 326

Step 3

		215, 216, 221	315, 326						
0	1	2	3	4	5	6	7	8	9

flatten:

215, 216, 221, 315, 326

final
answer

pseudocode

nums = [225, 315, 221, 326, 216]

$W = \text{len}(\text{strs}(\max(\text{nums})))$ # digits in max number

def radixsort(nums):

for digit in range(0, W): $O(W)$

$B = [[] \text{ for } x \text{ in range}(10)]$

for item in nums:

insert_index = item // $(10^{W-\text{digit}}) \% 10$

$O(\text{size of } B)$
array

$B[\text{insert_index}] \cdot \text{append}(\text{item})$

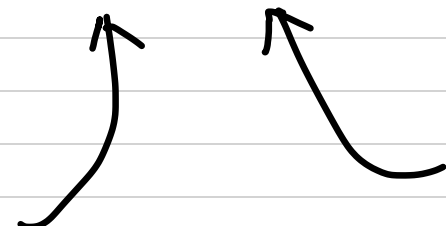
nums = flatten(B)

Time Complexity

$O(n \cdot w)$

Number of
values to
be sorted

length of longest
value

A diagram showing the time complexity $O(n \cdot w)$ with two arrows. One arrow points from the 'n' to the text 'Number of values to be sorted'. The other arrow points from the 'w' to the text 'length of longest value'.

