**6 kyu**

**N smallest elements in original order (performance edition)**

7284% of 1923 of38[FArekkusu](https://www.codewars.com/users/FArekkusu)

JavaScript

* [TRAIN AGAIN](https://www.codewars.com/kata/n-smallest-elements-in-original-order-performance-edition/train/javascript)
* [NEXT KATA](https://www.codewars.com/trainer/javascript)

Details

[Solutions](https://www.codewars.com/kata/n-smallest-elements-in-original-order-performance-edition/solutions/javascript)

[Forks (6)](https://www.codewars.com/kata/n-smallest-elements-in-original-order-performance-edition/forks/javascript)

[Discourse (32)](https://www.codewars.com/kata/n-smallest-elements-in-original-order-performance-edition/discuss/javascript)

* Add to Collection
* |
* Share this kata:

This challenge is based on [the kata](https://www.codewars.com/kata/n-smallest-elements-in-original-order" \t "_blank) by GiacomoSorbi. Before doing this one it is advisable to complete the non-performance version first.

**Task**

You will be given an array of random integers and a number n. You have to extract n smallest integers out of it **preserving the original order**.

**Examples**

performantSmallest([1, 2, 3, 4, 5], 3) === [1, 2, 3]

performantSmallest([5, 4, 3, 2, 1], 3) === [3, 2, 1]

performantSmallest([1, 2, 3, 4, 1], 3) === [1, 2, 1]

performantSmallest([1, 2, 3, -4, 0], 3) === [1, -4, 0]

performantSmallest([2, 1, 3, 2, 3], 3) === [2, 1, 2]

**Notes**

* The number n will always be smaller then the array length
* There will be duplicates in the array, and they have to be returned in the order of their each separate appearence
* Your solution has to be at least **as fast** as reference, passing the tests in 12000+ ms
* If you didn't pass a small amount of tests, try submitting again. Otherwise, your solution is not efficient enough.

**Test suite**

Tests: 1500

Array size: 9000

Numbers range: [-100..100]

Number of elements to return: 50-100% of the array

[**https://www.codewars.com/kata/n-smallest-elements-in-original-order-performance-edition/javascript**](https://www.codewars.com/kata/n-smallest-elements-in-original-order-performance-edition/javascript)

1. **function** performantSmallest(arr, n) {
3. **var** negativos = [];
4. **for**(let i =0; i<101; i++) negativos[i] = 0;
6. **var** positivos = [];
7. **for**(let i = 0; i<101; i++) positivos[i] = 0;
9. **for** (let i = 0; i < arr.length; i++)
10. {
11. **if** (arr[i] < 0)
12. {
13. negativos[-arr[i]]++;
14. }
15. **else**
16. {
17. positivos[arr[i]]++;
18. }
19. }
21. **var** ordenado = [];
22. **var** cont = 0;
23. **for** (let i = -100; i <= -1; i++)
24. {
25. **if** (negativos[-i] > 0)
26. {
27. **for** (let j = 0; j < negativos[-i]; j++)
28. {
29. ordenado.push(i);
30. cont++;
31. **if** (cont >= n)
32. {
33. **break**;
34. }
35. *//negativos[-i]--;*
36. }
37. **if** (cont >= n)
38. {
39. **break**;
40. }
41. }
42. }
43. **if** (cont < n)
44. {
45. **for** (let i = 0; i <= 100; i++)
46. {
47. **if** (positivos[i] > 0)
48. {
49. **for** (let j = 0; j < positivos[i]; j++)
50. {
51. ordenado.push(i);
52. cont++;
53. **if** (cont >= n)
54. {
55. **break**;
56. }
57. }
58. }
59. **if** (cont >= n)
60. {
61. **break**;
62. }
64. }
65. }

68. **var** negOrdenado = []; *// new int[101];*
69. **for**(let i =0; i<101; i++) negOrdenado[i] = 0;
70. **var** posOrdenado = []; *// new int[101];*
71. **for**(let i =0; i<101; i++) posOrdenado[i] = 0;
73. **for** (let i = 0; i < ordenado.length; i++)
74. {
75. **if** (ordenado[i] < 0)
76. {
77. negOrdenado[-ordenado[i]]++;
78. }
79. **else**
80. {
81. posOrdenado[ordenado[i]]++;
82. }
83. }

86. **var** ordTamN = [];
88. **for**(let i =0; i<arr.length; i++)
89. {
90. **if**(arr[i] < 0)
91. {
92. **if**(negOrdenado[-arr[i]] > 0)
93. {
94. ordTamN.push(arr[i]);
95. negOrdenado[-arr[i]]--;
96. }
97. }
98. **else**
99. {
100. **if**(posOrdenado[arr[i]] > 0)
101. {
102. ordTamN.push(arr[i]);
103. posOrdenado[arr[i]]--;
104. }
105. }
107. }
109. **return** ordTamN;
110. }

113. **var** arr = [ 1, 2, 3, -4, 0 ];
114. **var** n = 3;
115. print(performantSmallest(arr, n));