H

Full Name:

Đào Nguyễn Huy Hoàng

Email:

hoang.dnh210380@sis.hust.edu.vn

Test Name:

Mock Test

Taken On:

21 Apr 2023 21:50:52 IST

Time Taken:

13 min 57 sec/ 30 min

Invited by:

Ankush

Invited on:

21 Apr 2023 21:50:33 IST

Skills Score:

Tags Score:

Algorithms 95/95

Arrays 95/95

Core CS 95/95

Data Structures 95/95

Easy 95/95

Sorting 95/95

Strings 95/95

problem-solving 95/95

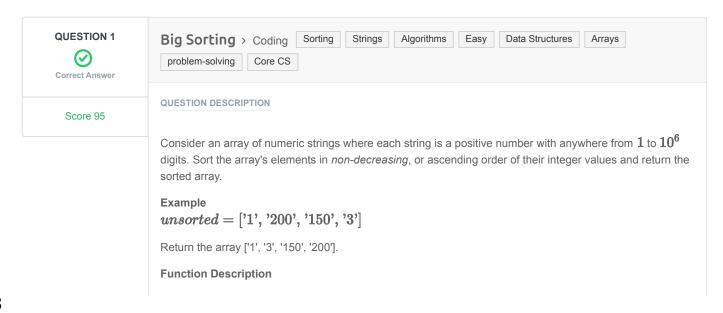
100% 95/95

scored in **Mock Test** in 13 min 57 sec on 21 Apr 2023 21:50:52 IST

Recruiter/Team Comments:

No Comments.

Question Description	Time Taken	Score	Status
Q1 Big Sorting > Coding	13 min 50 sec	95/ 95	Ø



Complete the bigSorting function in the editor below.

bigSorting has the following parameter(s):

• string unsorted[n]: an unsorted array of integers as strings

Returns

• string[n]: the array sorted in numerical order

Input Format

The first line contains an integer, n, the number of strings in unsorted. Each of the n subsequent lines contains an integer string, unsorted[i].

Constraints

- $1 \le n \le 2 \times 10^5$
- Each string is guaranteed to represent a positive integer.
- There will be no leading zeros.
- ullet The total number of digits across all strings in unsorted is between 1 and 10^6 (inclusive).

Sample Input 0

```
6
31415926535897932384626433832795
1
3
10
3
5
```

Sample Output 0

```
1
3
3
5
10
31415926535897932384626433832795
```

Explanation 0

The initial array of strings is

unsorted = [31415926535897932384626433832795, 1, 3, 10, 3, 5]. When we order each string by the real-world integer value it represents, we get:

$$1 \leq 3 \leq 3 \leq 5 \leq 10 \leq 31415926535897932384626433832795$$

We then print each value on a new line, from smallest to largest.

Sample Input 1

```
8
1
2
100
12303479849857341718340192371
3084193741082937
3084193741082938
111
200
```

Sample Output 1

```
1
2
100
```

```
111
200
3084193741082937
3084193741082938
12303479849857341718340192371
```

CANDIDATE ANSWER

Language used: Java 7

```
class Result implements Comparator<String>{

public int compare(String str1, String str2) {
    return (str1.length()==str2.length()) ? str1.compareTo(str2) :

str1.length()-str2.length();
}

public static List<String> bigSorting(List<String> unsorted) {
    Collections.sort(unsorted, new Result());
    return unsorted;
}
```

TESTCASE	DIFFICULTY	TYPE	STATUS	SCORE	TIME TAKEN	MEMORY USED
Testcase 1	Easy	Sample case	Success	0	0.074 sec	21.5 KB
Testcase 2	Medium	Hidden case	Success	10	0.1231 sec	21.6 KB
Testcase 3	Medium	Hidden case	Success	10	0.1 sec	26.3 KB
Testcase 4	Hard	Hidden case	Success	15	0.2184 sec	32.4 KB
Testcase 5	Hard	Hidden case	Success	15	0.1357 sec	34 KB
Testcase 6	Hard	Hidden case	Success	15	0.1388 sec	30.6 KB
Testcase 7	Hard	Hidden case	Success	15	0.2151 sec	35.9 KB
Testcase 8	Hard	Hidden case	Success	15	0.2474 sec	42.9 KB
Testcase 9	Easy	Sample case	Success	0	0.0579 sec	21.4 KB

No Comments

PDF generated at: 21 Apr 2023 16:36:33 UTC