(/problemset/all/)   Contest(/contest/)   Discuss(/discuss/)   Interview ⌄   Store ⌄

(/problemset/all/)

(/subscribe?ref=nb_npl&source=nav-icon)
◇ 0 subm
rearra

🔔 ⚡

SALE   THANKS2023   USE NOW ❯
(/subscribe?ref=nb_npl&source=site-announcement) ✕

100142. Make Lexicographically Smallest Array by Swapping Elements

My Submissions (/contest/weekly-contest-373/problems/make-lexicographically-smallest-array-by-swapping-elements/submissions/)

Back to Contest (/contest/weekly-contest-373/)

You are given a **0-indexed** array of **positive** integers `nums` and a **positive** integer `limit`.

In one operation, you can choose any two indices `i` and `j` and swap `nums[i]` and `nums[j]` **if** $|nums[i] - nums[j]| <= limit$.

Return *the **lexicographically smallest array** that can be obtained by performing the operation any number of times.*

An array `a` is lexicographically smaller than an array `b` if in the first position where `a` and `b` differ, array `a` has an element that is less than the corresponding element in `b`. For example, the array `[2,10,3]` is lexicographically smaller than the array `[10,2,3]` because they differ at index `0` and `2 < 10`.

User Accepted:	926
User Tried:	2196
Total Accepted:	940
Total Submissions:	3575
Difficulty:	Medium

Example 1:

```
Input: nums = [1,5,3,9,8], limit = 2
Output: [1,3,5,8,9]
Explanation: Apply the operation 2 times:
- Swap nums[1] with nums[2]. The array becomes [1,3,5,9,8]
- Swap nums[3] with nums[4]. The array becomes [1,3,5,8,9]
We cannot obtain a lexicographically smaller array by applying any more operations.
Note that it may be possible to get the same result by doing different operations.
```

Example 2:

```
Input: nums = [1,7,6,18,2,1], limit = 3
Output: [1,6,7,18,1,2]
Explanation: Apply the operation 3 times:
- Swap nums[1] with nums[2]. The array becomes [1,6,7,18,2,1]
- Swap nums[0] with nums[4]. The array becomes [2,6,7,18,1,1]
- Swap nums[0] with nums[5]. The array becomes [1,6,7,18,1,2]
We cannot obtain a lexicographically smaller array by applying any more operations.
```

Example 3:

```
Input: nums = [1,7,28,19,10], limit = 3
Output: [1,7,28,19,10]
Explanation: [1,7,28,19,10] is the lexicographically smallest array we can obtain because we cannot apply the operation on any
```

Constraints:

- $1 <= nums.length <= 10^5$
- $1 <= nums[i] <= 10^9$
- $1 <= limit <= 10^9$

JavaScript ⌄   ⟨/⟩ ⟳ ⚙

```javascript
 1  function DJSet(n) {
 2      let parent = Array(n).fill(-1);
 3      return { find, union, count, equiv, par, grp }
 4      function find(x) {
 5          return parent[x] < 0 ? x : parent[x] = find(parent[x]);
 6      }
 7      function union(x, y) {
 8          x = find(x);
 9          y = find(y);
10          if (x == y) return false;
11          if (parent[x] < parent[y]) [x, y] = [y, x];
```

```
12              parent[x] += parent[y];
13              parent[y] = x;
14              return true;
15          }
16 ▾     function count() { // total groups
17              return parent.filter(v => v < 0).length;
18          }
19 ▾     function equiv(x, y) { // isConnected
20              return find(x) == find(y);
21          }
22 ▾     function par() {
23              return parent;
24          }
25 ▾     function grp() {
26              let groups = [];
27              for (let i = 0; i < n; i++) groups.push([]);
28              for (let i = 0; i < n; i++) groups[find(i)].push(i); // sorted and unique
29              return groups;
30          }
31  }
32
33  // https://leetcode.com/problems/smallest-string-with-swaps/
34 ▾ const LexicalSmallestArrayWithSwaps = (a, pairs) => {
35      let n = a.length, ds = new DJSet(n), res = Array(n).fill(0);
36      for (const [x, y] of pairs) ds.union(x, y);
37      let groups = ds.grp().filter(e => e.length);
38 ▾     for (const group of groups) {
39          let ga = [];
40          for (let i of group) ga.push(a[i]);
41          ga.sort((x, y) => x - y);
42          for (let i = 0; i < group.length; i++) res[group[i]] = ga[i];
43      }
44      return res;
45  };
46
47 ▾ const lexicographicallySmallestArray = (a, limit) => {
48      let d = a.map((x, i) => [x, i]).sort((x, y) => x[0] - y[0] || x[1] - y[1]), pairs = [];
49 ▾     for (let i = 1; i < a.length; i++) {
50          if (d[i][0] - d[i - 1][0] <= limit) pairs.push([d[i - 1][1], d[i][1]]);
51      }
52      return LexicalSmallestArrayWithSwaps(a, pairs)
53  };
```

☐ **Custom Testcase**   (Use Example Testcases)

(▶ Run)   (☁ Submit)

Submission Result: *Accepted* (/submissions/detail/1106544248/) ❓   (More Details ❯ (/submissions/detail/1106544248/))

Share your acceptance!