

ref=nb_npl)





5860. Find Original Array From Doubled Array

My Submissions (/contest/biweekly-contest-61/problems/find-original-array-from-doubled-array/submissions/)

Back to Contest (/contest/biweekly-contest-61/)

An integer array original is transformed into a doubled array changed by appending twice the value of every element in original, and then randomly shuffling the resulting array.

Given an array changed, return original if changed is a doubled array. If changed is not a doubled array, return an empty array. The elements in original may be returned in any order.

User Accepted: 0 **User Tried:** 0 **Total Accepted:** 0 **Total Submissions:** 0 Medium Difficulty:

Example 1:

```
Input: changed = [1,3,4,2,6,8]
Output: [1,3,4]
Explanation: One possible original array could be [1,3,4]:
- Twice the value of 1 is 1 * 2 = 2.
- Twice the value of 3 is 3 * 2 = 6.
- Twice the value of 4 is 4 * 2 = 8.
Other original arrays could be [4,3,1] or [3,1,4].
```

Example 2:

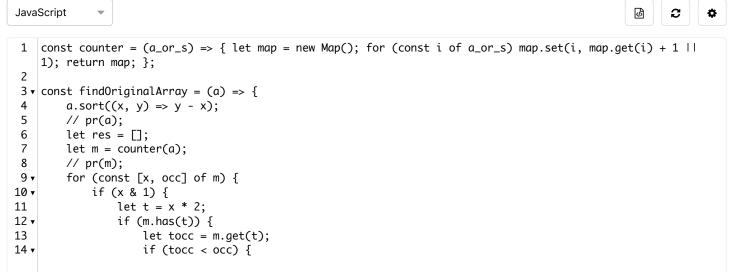
```
Input: changed = [6,3,0,1]
Output: []
Explanation: changed is not a doubled array.
```

Example 3:

```
Input: changed = [1]
Output: []
Explanation: changed is not a doubled array.
```

Constraints:

- 1 <= changed.length <= 10⁵
- 0 <= changed[i] <= 10⁵



```
9/18/21, 11:48 AM
                                                                                                                                               Find Original Array From Doubled Array - LeetCode Contest
         15
                                                                                   return [];
          16 ▼
                                                                      } else if (tocc == occ) {
                                                                                   m.delete(x)
          17
          18
                                                                                   m.delete(t)
          19
                                                                                   let repeat = occ;
          20
                                                                                   while (repeat--) res.push(x);
                                                                      } else {
          21 •
          22
                                                                                   m.delete(x)
          23
                                                                                   m.set(t, tocc - occ);
          24
                                                                                   let repeat = occ;
          25
                                                                                   while (repeat--) res.push(x);
          26
                                                                       }
          27 •
                                                           } else {
          28
                                                                       return [];
          29
          30
                                              }
          31
          32
                                  // pr(m);
          33 ▼
                                  for (const [x, occ] of m) {
          34 ▼
                                              if (x == 0) {
          35
                                                           if (occ & 1) return [];
          36
                                                           let repeat = occ / 2;
          37
                                                           while (repeat--) res.push(x);
          38
                                                           m.delete(x);
          39
                                                           continue;
          40
                                               }
          41
                                              let t = x / 2;
          42 v
                                               if (m.has(t)) {
          43
                                                           let tocc = m.get(t);
                                                           if (tocc < occ) {
          44 ▼
          45
                                                                       return [];
                                                           } else if (tocc == occ) {
          46 ▼
          47
                                                                      m.delete(x)
          48
                                                                      m.delete(t)
          49
                                                                       let repeat = occ;
          50
                                                                      while (repeat--) res.push(t);
          51 ▼
                                                           } else {
          52
                                                                      m.delete(x)
          53
                                                                      m.set(t, tocc - occ);
          54
                                                                      let repeat = occ;
          55
                                                                      while (repeat--) res.push(t);
          56
                                                           }
          57 ▼
                                               } else {
          58
                                                           return [];
          59
                                              }
          60
          61
                                  // pr(m);
          62
                                   return m.size > 0 ? [] : res;
          63
                      };
      ☐ Custom Testcase
                                                                   Use Example Testcases
                                                                                                                                                                                                                                                                                                     Run

    Submit
    Su
      Submission Result: Accepted (/submissions/detail/557013610/) ?
                                                                                                                                                                                                                       More Details > (/submissions/detail/557013610/)
```

```
Share your acceptance!
Copyright © 2021 LeetCode
Help Center (/support) | Jobs (/jobs) | Bug Bounty (/bugbounty) | Online Interview (/interview/) | Students (/student) | Terms (/terms)
```

United States (/region)

Privacy Policy (/privacy)