6039. Maximum Product After K Increments

My Submissions (/contest/weekly-contest-288/problems/maximum-product-after-k-increments/submissions/) Back to Contest (/contest/weekly-contest-288/)

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

You are given an array of non-negative integers `nums` and an integer `k`. In one operation, you may choose **any** element from `nums` and **increment** it by `1`.

Return *the **maximum product** of* `nums` *after **at most*** `k` *operations*. Since the answer may be very large, return it **modulo** $10^9 + 7$.

Example 1:

```
Input: nums = [0,4], k = 5
Output: 20
Explanation: Increment the first number 5 times.
Now nums = [5, 4], with a product of 5 * 4 = 20.
It can be shown that 20 is maximum product possible, so we return 20.
Note that there may be other ways to increment nums to have the maximum product.
```

Example 2:

```
Input: nums = [6,3,3,2], k = 2
Output: 216
Explanation: Increment the second number 1 time and increment the fourth number 1 time.
Now nums = [6, 4, 3, 3], with a product of 6 * 4 * 3 * 3 = 216.
It can be shown that 216 is maximum product possible, so we return 216.
Note that there may be other ways to increment nums to have the maximum product.
```

Constraints:

- $1 <= nums.length, k <= 10^5$
- $0 <= nums[i] <= 10^6$

Java

```java
class Solution {
    private final int mod = (int) 1e9 + 7;

    public int maximumProduct(int[] a, int k) {
        TreeMap<Integer, Integer> m = counter(a);
        while (k-- > 0) {
            int min = m.firstKey();
            addOneOrManyMap(m, min + 1);
            removeOneOrManyMap(m, min);
        }
        // tr(m);
        long res = 1;
        for (int x : m.keySet()) {
            res = (res * pow_mod(x, m.get(x))) % mod;
        }
        return (int) res;
    }

    long pow_mod(long a, long b) {
        long r = 1;
        while (b > 0) {
            if (b % 2 == 1) r = r * a % mod;
            b >>= 1;
            a = a * a % mod;
        }
        return r;
    }

    <T> void addOneOrManyMap(TreeMap<T, Integer> m, T x, int... args) {
        int cnt = args.length == 0 ? 1 : args[0];
        m.put(x, m.getOrDefault(x, 0) + cnt);
    }

    <T> void removeOneOrManyMap(TreeMap<T, Integer> m, T x, int... args) {
        int cnt = args.length == 0 ? 1 : args[0], occ = m.get(x);
        if (occ > cnt) {
            m.put(x, occ - cnt);
        } else {
            m.remove(x);
        }
    }

    TreeMap<Integer, Integer> counter(int[] a) {
        TreeMap<Integer, Integer> m = new TreeMap<>();
        for (int x : a) m.put(x, m.getOrDefault(x, 0) + 1);
        return m;
    }
}
```

□ **Custom Testcase** Use Example Testcases

Run Submit

Submission Result: Accepted (/submissions/detail/677376790/) More Details (/submissions/detail/677376790/)

Share your acceptance!

◀ 1

Help Center (/support) | Jobs (/jobs) | Bug Bounty (/bugbounty) | Online Interview (/interview/) | Students (/student) | Terms (/terms) | Privacy Policy (/privacy)

United States (/region)