5425. Maximum Area of a Piece of Cake After Horizontal and Vertical Cuts

/weekly-contest-191/problems/maximum-area-of-a-piece-of-cake-after-horizontal-and-vertical-cuts/submissions/)

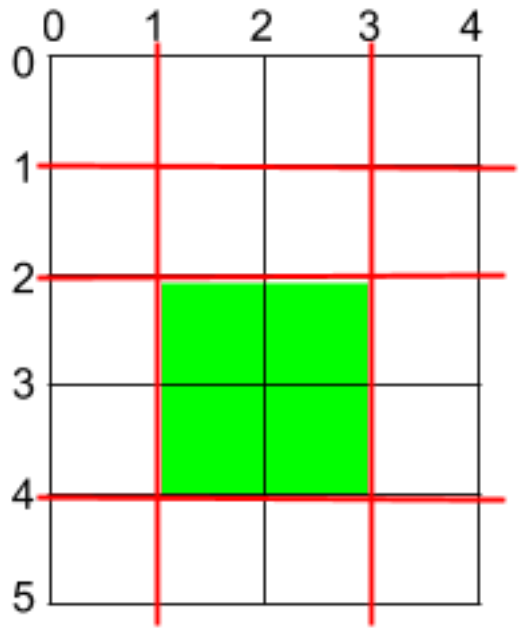
'weekly-contest-191/)

Given a rectangular cake with height `h` and width `w`, and two arrays of integers `horizontalCuts` and `verticalCuts` where `horizontalCuts[i]` is the distance from the top of the rectangular cake to the `ith` horizontal cut and similarly, `verticalCuts[j]` is the distance from the left of the rectangular cake to the `jth` vertical cut.

Return the maximum area of a piece of cake after you cut at each horizontal and vertical position provided in the arrays `horizontalCuts` and `verticalCuts`. Since the answer can be a huge number, return this modulo 10^9 + 7.

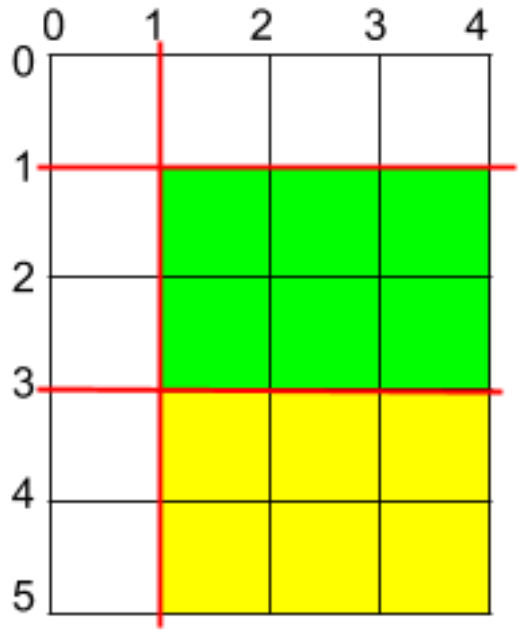
User Accepted:	4577
User Tried:	6033
Total Accepted:	4644
Total Submissions:	17396
Difficulty:	Medium

Example 1:




```
Input: h = 5, w = 4, horizontalCuts = [1,2,4], verticalCuts = [1,3]
Output: 4
Explanation: The figure above represents the given rectangular cake. Red lines are the hor
```

Example 2:




```
Input: h = 5, w = 4, horizontalCuts = [3,1], verticalCuts = [1]
Output: 6
Explanation: The figure above represents the given rectangular cake. Red lines are the hor
```

Example 3:

```
Input: h = 5, w = 4, horizontalCuts = [3], verticalCuts = [3]
Output: 9
```

Constraints:

- `2 <= h, w <= 10^9`
- `1 <= horizontalCuts.length < min(h, 10^5)`
- `1 <= verticalCuts.length < min(w, 10^5)`
- `1 <= horizontalCuts[i] < h`
- `1 <= verticalCuts[i] < w`
- It is guaranteed that all elements in `horizontalCuts` are distinct.
- It is guaranteed that all elements in `verticalCuts` are distinct.

JavaScript

```
/**
 * @param {number} h
 * @param {number} w
 * @param {number[]} horizontalCuts
 * @param {number[]} verticalCuts
 * @return {number}
 */
const maxArea = (h, w, horizontalCuts, verticalCuts) => {
    let x = Number.MIN_VALUE;
```

```
    let y = Number.MIN_VALUE;
    horizontalCuts.sort((a, b) => b - a);
    verticalCuts.sort((a, b) => b - a);
    for (let i = 1; i < horizontalCuts.length; i++) {
        x = Math.max(Math.abs(horizontalCuts[i - 1] - horizontalCuts[i]), x);
    }
    x = Math.max(Math.abs(h - horizontalCuts[0]), x);
    x = Math.max(Math.abs(horizontalCuts[horizontalCuts.length - 1] - 0), x);

    for (let i = 1; i < verticalCuts.length; i++) {
        y = Math.max(Math.abs(verticalCuts[i - 1] - verticalCuts[i]), y);
    }
    y = Math.max(Math.abs(w - verticalCuts[0]), y);
    y = Math.max(Math.abs(verticalCuts[verticalCuts.length - 1] - 0), y);

    return x * y;
};
```

☐ **Custom Testcase** Use Example Testcases

Run Submit

Submission Result: Wrong Answer (/submissions/detail/346975040/)

More Details ❯ (/submissions/detail/346975040/)

Input: **1000000000**
1000000000
[591015057,176484069,195962145,951546667,439279063,998302483,659…
[792989830,499549413,794617178,574089005,411436182,921342115,946…

Output: **13755333066**

Expected: **755332975**