Problem 2943: Maximize Area of Square Hole in Grid

Problem Information

Difficulty: Medium
Acceptance Rate: 37.57%
Paid Only: No
Tags: Array, Sorting

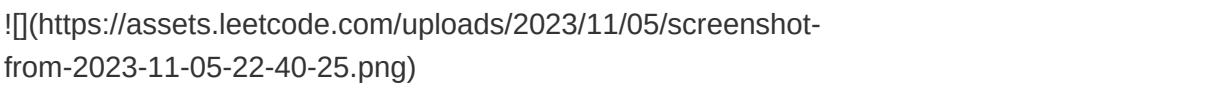
Problem Description

You are given the two integers, `n` and `m` and two integer arrays, `hBars` and `vBars`. The grid has `n + 2` horizontal and `m + 2` vertical bars, creating 1 x 1 unit cells. The bars are indexed starting from `1`.

You can **remove** some of the bars in `hBars` from horizontal bars and some of the bars in `vBars` from vertical bars. Note that other bars are fixed and cannot be removed.

Return an integer denoting the **maximum area** of a _square-shaped_ hole in the grid, after removing some bars (possibly none).

Example 1:



Input: n = 2, m = 1, hBars = [2,3], vBars = [2]

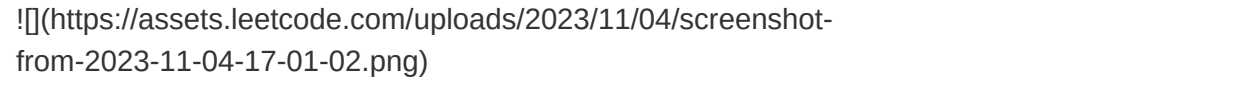
Output: 4

Explanation:

The left image shows the initial grid formed by the bars. The horizontal bars are `[1,2,3,4]`, and the vertical bars are `[1,2,3]`.

One way to get the maximum square-shaped hole is by removing horizontal bar 2 and vertical bar 2.

Example 2:



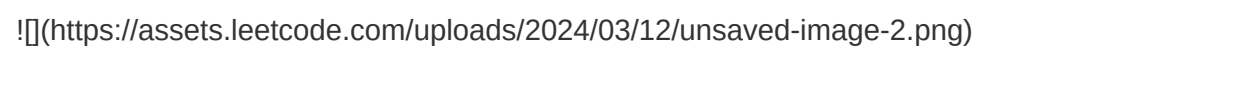
Input: n = 1, m = 1, hBars = [2], vBars = [2]

Output: 4

Explanation:

To get the maximum square-shaped hole, we remove horizontal bar 2 and vertical bar 2.

Example 3:



Input: n = 2, m = 3, hBars = [2,3], vBars = [2,4]

Output: 4

Explanation:

One way to get the maximum square-shaped hole is by removing horizontal bar 3, and vertical bar 4.

Constraints:

* `1 <= n <= 109` * `1 <= m <= 109` * `1 <= hBars.length <= 100` * `2 <= hBars[i] <= n + 1` * `1 <= vBars.length <= 100` * `2 <= vBars[i] <= m + 1` * All values in `hBars` are distinct. * All values in `vBars` are distinct.

Code Snippets

C++:

```
class Solution {
public:
int maximizeSquareHoleArea(int n, int m, vector<int>& hBars, vector<int>&
vBars) {

}
};
```

Java:

```
class Solution {
public int maximizeSquareHoleArea(int n, int m, int[] hBars, int[] vBars) {

}
}
```

Python3:

```
class Solution:
def maximizeSquareHoleArea(self, n: int, m: int, hBars: List[int], vBars:
List[int]) -> int:
```