2418. Sort The People

Easy

Topics

Companies

Hint

You are given an array of strings `names`, and an array `heights` that consists of **distinct** positive integers. Both arrays are of length `n`.

For each index `i`, `names[i]` and `heights[i]` denote the name and height of the `ith` person.

Return `names` *sorted in* ***descending*** *order by the people's heights*.

Example 1:

Input: names = ["Mary","John","Emma"], heights = [180,165,170]
Output: ["Mary","Emma","John"]
Explanation: Mary is the tallest, followed by Emma and John.

Example 2:

Input: names = ["Alice","Bob","Bob"], heights = [155,185,150]
Output: ["Bob","Alice","Bob"]
Explanation: The first Bob is the tallest, followed by Alice and the second Bob.

Constraints:

- `n == names.length == heights.length`
- `1 <= n <= 103`
- `1 <= names[i].length <= 20`
- `1 <= heights[i] <= 105`
- `names[i]` consists of lower and upper case English letters.
- All the values of `heights` are distinct.

Solution:

```
class Solution {

    public String[] sortPeople(String[] names, int[] heights) {

        for (int i = 0; i < heights.length - 1; i++) {
            // Find the index of the maximum height in the remaining array
            int maxIndex = i;
            for (int j = i + 1; j < heights.length; j++) {
                if (heights[j] > heights[maxIndex]) {
                    maxIndex = j;
                }
            }
            // Swap the heights
            int tempHeight = heights[i];
            heights[i] = heights[maxIndex];
            heights[maxIndex] = tempHeight;

            // Swap the corresponding names
            String tempName = names[i];
            names[i] = names[maxIndex];
            names[maxIndex] = tempName;
        }
        return names;
    }
}
```