2418. Sort the People

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 `i`th 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.
```

```
/*
    construct an array over pair (height, name)
    Then sort it wrt greater.
    Reuse names to return the answer.
   Time complexity: O(n+nlogn+n)=O(nlogn)
   Extra space complexity: O(2n+logn)
*/
class Solution {
public:
    std::vector<std::string> sortPeople(std::vector<string>& names, std::vector<int>& heights) {
        std::vector<std::pair<int,std::string>> heights_names;
        int n=names.size();
        for(int i=0;i<n;++i) heights_names.push_back({heights[i],names[i]});
        std::sort(heights_names.begin(),heights_names.end(),std::greater<>());
        for(int i=0;i<n;++i) names[i]=heights_names[i].second;
        return names;
    }
};
```

2418. Sort the People

```
/*
    Pack each height with its correspondant index (height[i],i)
    sort the packed data (based on height)
    unpack tbe data, ino order to put each name in unpakded i, in its correct position

    Time complexity: O(n+nlogn+n)=O(nlogn)
    Extra space complexity: O(n+logn)
*/
class Solution {
public:
    std::vector<std::string> sortPeople(std::vector<string>& names, std::vector<int>& heights) {
        const int n=names.size();
        for(int i=0; i<n; i++) heights[i]=((heights[i]<<10)|i);

        sort(heights.begin(), heights.end(), greater<>());

        vector<string> ans(n);
        for(int i=0; i<n; i++) ans[i]=names[heights[i]&1023];
        return ans;
    }
};
```