

```

class Solution {
    vector<int> priority;
public:
    Solution(){
        for (int i = 0; i < 26; i++)
            priority.push_back(25);
    }

    void updatePriority(string order){
        int k = 1;
        for (int i = 0; i < order.size(); i++){
            priority[order[i] - 'a'] = i;
        }
    }

    void insertInSorted(string& s, int i){
        int j = i - 1;
        while (j >= 0 && priority[s[j] - 'a'] > priority[s[i] - 'a']){
            swap(s[i], s[j]);
            i--;
            j--;
        }
    }

    void insertionSort(string& s){
        for (int i = 1; i < s.size(); i++)
            insertInSorted(s, i);
    }

    string customSortString(string order, string s) {
        updatePriority(order);
        for (auto x: priority)
            cout << x << " ";

        insertionSort(s);

        return s;
    }
};

```