

给定一个仅包含小写字母的字符串，去除字符串中的重复字符，使得每个字符重复出现一次，需保证结果的节点序最小。

分析：

1. 针对去重，采用`vector<bool>`标记，表示当前元素是否被访问过；

```
class Solution {
public:
    std::string removeDuplicateLetters(std::string s) {
        std::list<char> st;
        std::map<int, int> map;
        std::vector<bool> visited(256, false);
        for (int i = 0; i < s.size(); i++) {
            map[s[i]]++;
        }

        for (int i = 0; i < s.size(); i++) {
            char ch = s[i];
            map[ch]--;

            if (visited[ch]) {
                continue; // 已访问过
            }

            while (!st.empty() && st.back() > ch) {
                if (map[st.back()] == 0) {
                    break; // 后续不存在相关字符，终止
                }
                visited[st.back()] = false;
                st.pop_back();
            }

            st.push_back(ch);
            visited[ch] = true;
        }

        std::string res;
        for (auto item : st) {
            res += item;
        }

        return res;
    }
};
```

已排序数组去重，保留k个

```
class Solution{
public:
    // 保留k个
    int removeDuplicate(std::vector<int> &nums, int k){
        int len = nums.size();
        if(len < k ) {
            return len;
        }

        int pos = k;
        for(int i = k;i<len;i++) {
            if(nums[pos-k] != nums[i]){
                nums[pos++] = nums[i];
            }
        }

        return pos;
    }
};
```