

和谐数组是指一个数组里元素的最大值和最小值之间的差别正好是1。

现在，给定一个整数数组，你需要在所有可能的子序列中找到最长的和谐子序列的长度。

示例 1:

输入: [1, 3, 2, 2, 5, 2, 3, 7]

输出: 5

原因: 最长的和谐数组是: [3, 2, 2, 2, 3] .

说明: 输入的数组长度最大不超过20,000.

需注意，改题中最大元素和最小元素差正好为1，那么该数组中，必包含两个元素，他们间的差值为1，通过这个条件，在动态的计算每个元素出现的值（c++map好啊，没出现的默认为0，避免了未包含）

```
class Solution {
public:
    int findLHS(vector<int>& nums) {
        map<int,int> bj;
        int max=0;
        for(int i=0;i<nums.size();i++){
            bj[nums[i]]++;
            if(bj[nums[i]]!=0&&bj[nums[i]-1]!=0){
                max=max>=bj[nums[i]]+bj[nums[i]-1]?max:bj[nums[i]]+bj[nums[i]-1];
            }
            if(bj[nums[i]]!=0&&bj[nums[i]+1]!=0){
                max=max>=bj[nums[i]]+bj[nums[i]+1]?max:bj[nums[i]]+bj[nums[i]+1];
            }
        }
        return max;
    }
};
```