2.Medium\10.Longest_consecutive_subsequence.cpp

```cpp
/*
QUESTION:-

Given an unsorted array of integers nums, return the length of the longest consecutive elements sequence.

Example 1:
Input: nums = [100,4,200,1,3,2]
Output: 4
Explanation: The longest consecutive elements sequence is [1, 2, 3, 4]. Therefore its length is 4.

Example 2:
Input: nums = [0,3,7,2,5,8,4,6,0,1]
Output: 9
*/

/*
APPROACH:-

To find the length of the longest consecutive elements sequence, we can follow these steps:

1. Create a set to store all the elements of the array.
2. Iterate through the array and insert each element into the set.
3. For each element, check if its previous consecutive element (num-1) exists in the set. If it does not exist, it means the current element is the starting element of a sequence.
4. For each starting element, keep incrementing the current element (num+1) and checking if it exists in the set. This will help find the consecutive elements in the sequence.
5. Keep track of the maximum length of consecutive elements encountered.
6. Return the maximum length as the result.

*/

// CODE:
int longestConsecutive(vector<int> &nums)
{
    unordered_map<int, int> mp;
    for (int i = 0; i < nums.size(); i++)
    {
        mp[nums[i]]++;
    }
    int ans = 0;
    for (int i = 0; i < nums.size(); i++)
    {
        int start = nums[i];
        // check whehter this can be the start of the subsequence
        if (mp.find(nums[i] - 1) == mp.end())
        {
            int temp = 1;
            int nxt = nums[i];
            while (mp.find(nxt + 1) != mp.end())
            {
                temp++;
                nxt++;
            }
            ans = max(ans, temp);
        }
    }
    return ans;
}

// TIME COMPLEXITY: O(n), where n is the size of the input array.
// SPACE COMPLEXITY: O(n), as we are using a set to store the elements of the array.

```