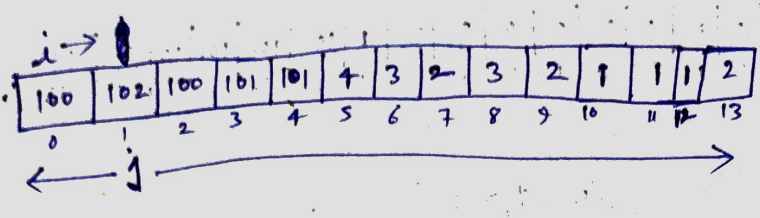
Longest Consecutive Sequence

↳ Same as "Longest Subsequence of consecutive elements"



- **Brute Force :-**

1) i=0
current = 100 // arr[i]
tempLength = 1 maxLength = $-\infty$

2) Find 101 in the entire array
↳ Found. So, current = 101
tempLength = 2

3) Find 102 in the entire array.
↳ Found. So, current = 102
tempLength = 3

4) Find 103 in the entire array.
↳ Not Found. So, if (maxLength < tempLength) {
maxLength = tempLength;
}
i++;

5) i=1
current = 102 // arr[i]
tempLength = 1

6) Find 103 in the entire array.
↳ Not Found. if (maxLen > tempLen) {
maxLen = tempLen;
}
i++;

7) i=2
current = 100
tempLength = 1

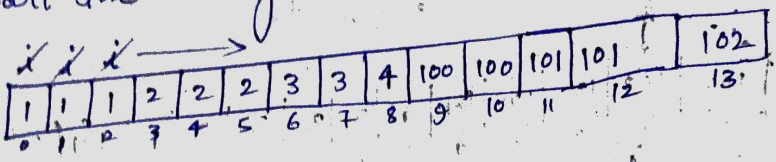
.....

So on....

Here, we are iterating over the array and finding the longest consecutive sequence starting with arr[i] and keep on adding its length. At last whatever the maximum length we have that is our answer. $T = O(n) * O(n^2) = O(n^3)$

• Better Solution :-

Sort the array :-



current = 1
templen = 1 maxLen = -∞

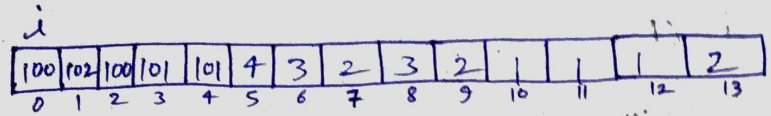
~~repeat (i=0; i++~~

```
for(i=0; i<=n-1; i++){
    if(arr[i] == current+1){
        current = current+1
        templen++;
    } else if(arr[i] == current){
        continue;
    } else {
        if(maxLen < templen){
            maxLen = templen;
        }
        current = arr[i];
        templen = 1;
    }
}
if(maxLen < templen){
    maxLen = templen;
}
```

$T = O(n * \log n) + O(n)$

$S = O(1)$

- **Best Solution :-**



1) Iterate and put elements in hashMap

2) Then, let us iterate over the array again with i=0.

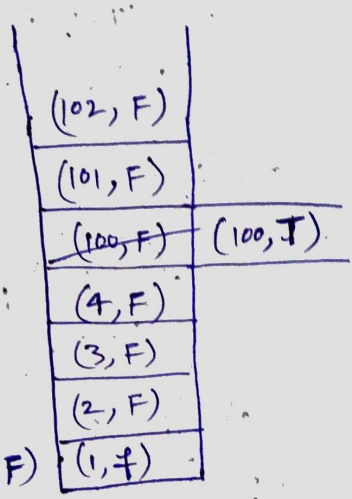
- if ((100-1) present in map)
  - ↳ No ⇒ Then 100 is starting point ⇒ (100, F)
  - ↳ Then check

100 → 101 → 102 → (103) X

Now, put(100, true) back in map.

i++

3
manLen



HashMap (Integer, Boolean)

Eg.

(100, false)

It represents length of sequence starting with 100 is not yet found.

3) i=1

if((102-1) is present in map)

↳ Yes ⇒ i++

4) i=2

if((100-1) is present in map)

↳ NO ⇒ Then 100 is starting point ⇒ But (100, true)

↳ i++

5) i=3

if ((101-1) present in map)

↳ Yes ⇒ i++

6) i=4

if ((101-1) present in map)

↳ Yes ⇒ i++

7) i=5

if ((4-1) present in map)

↳ Yes ⇒ i++

8) i=6

Similarly

12) i=10

if ((1-1) present in map)

↳ No ⇒ Then 1 is starting point

⇓

(1, F)

⇓

Then check

1 → 2 → 3 → 4 → X(5)

4
maxLen

Then put (1, true) back in the map

i++

13) $i = 11$

if((11-1) is present in map)

↳ No ⇒ Then 11 is starting point

⇓

(11, T)

⇓

i++

14) $i = 12$

Same as (13)

15) $i = 13$

if((13-1) is present in map)

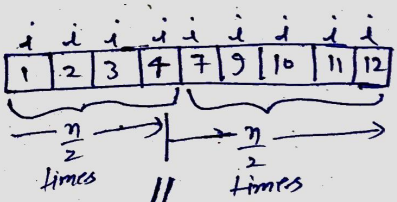
↳ Yes ⇒ i++

So, maxLength = 4

$T = O(n) + O(2n) = O(3n) = O(n)$

⇓

You have to think logically



⇓

$O(n) + O(\frac{n}{2}) + O(\frac{n}{2})$

$= O(n) + O(n)$

$= O(2n)$

So, $T = O(n)$ $S = O(n)$