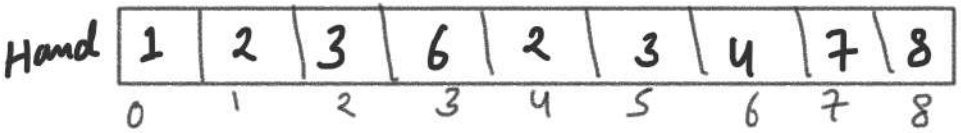
PROBLEM Hand of straights [Leetcode - 48]

EX1



groupsize = 3

[1 2 3]
[2 3 4]
[6 7 8]
} output TRUE

↳ Note

Cards group consecutive hona chaiye

it means currNumber = 1

[1, (1+1), (1+1+1)]

EX2

Hand

1	2	3	6	2	3	3	7	8
0	1	2	3	4	5	6	7	8

groupsize = 3

[1,2,3], [2,3,3]

X This is not consecutive group

Output : False

EX3

Hand

1	2	3	4	5

groupsize = 4

Group1 [1 2 3 4]

Group2 [5]

X This is not valid group due to groupsize

Ex1

Hand

1	2	3	6	2	3	4	7	8
0	1	2	3	4	5	6	7	8

groupsize = 3

[1 2 3]
[2 3 4]
[6 7 8]
} output TRUE

Approach

STEP1 Store the freqency of each value of hands in ordered map

map<int, int> mp;

key	value
1	1
2	2
3	2
4	1
6	1
7	1
8	1

STEP2 Create the consecutive cards group of groupsize

STEP2

Key	value
1	0
2	2 X 0
3	3 X 0
4	4 0
5	5 0
6	6 0
7	7 0
8	8 0

currNO = 1 if(map[curr] >= 1)

↳ group[1,2,3]

currNO = 2 . if(map[curr] >= 1)

↳ group[2,3,4]

currNO = 6 if(map[curr] >= 1)

↳ group[6,7,8] → TRUE

EX2

Hand

1	2	3	6	2	3	3	7	8
0	1	2	3	4	5	6	7	8

groupsize = 3

Approach

STEP1

STEP2

jump 1

[1, 2, 3]

CURR NO = 1

CURR NO = 2

[2, 3, 4]

Key	value
1	0
2	1
3	2
6	1
7	1
8	1

4 not available in map

it means we can not build

full group consecutively

Output = False

```cpp
class Solution {
public:
    bool isNStraightHand(vector<int>& hand, int groupSize) {
        // Edge Case
        int n = hand.size();
        if(n % groupSize != 0){
            return false;
        }

        // Step 1: tore the frequency of each value of hands in ordered map
        map<int, int> freqMap;
        for(auto &handValue: hand){
            freqMap[handValue]++;
        }

        // Step 2: create the consecutive cards group of group size till map is not empty
        while(!freqMap.empty()){
            int currNumber = freqMap.begin()->first;
            for(int i =0; i < groupSize; i++){
                if(freqMap[currNumber + i] == 0){
                    // Not present the consecutive number like: currNumber + 0, currNumber + 1
...
                    // it means not possible to build the consecutive cards group
                    return false;
                }

                // Update the frequency of currNumber
                freqMap[currNumber + i]--;

                if(freqMap[currNumber + i] < 1){
                    // currNumber + i frequency is now zero or -ve
                    // delete this number from map
                    freqMap.erase(currNumber + i);
                }
            }
        }

        // Agar ynha tak pahunch gya hu iska mtlb consecutive cards group ban gya hai
        return true;
    }
};
```

EX 1 Hand size = 9, groupsize = 3, 9 % 3 == 0 → True

EX 3 Hand size = 5, groupsize = 4, 5 % 4 != 0 → False

Time Complexity ⇒ O(N log N)

- STEP 1 O(N log N)
- STEP 2 O(N * groupsize)

S.C ⇒ O(N)