# APPROACH

In this approach, we will keep track of the most frequent element using a HashMap. HashMap will contain key-value pairs where the key will be the distinct element of the ‘ARR’ and value will be its frequency. Now, suppose X is the most frequent element. In that case, the special subarray will look like {X, …, X} because we can just remove any other preceding or succeeding element without affecting our answer as we want our special subarray to be the smallest possible. To obtain this subarray, we can just find the index of the first and the last occurrence of the most frequent element. We also have to keep track of the size of the subarrays for the case when the ‘ARR’ contains more than one element with maximum frequency.

Algorithm:

- Initialize a map 'LEFT' to store left most occurrence of every element.
- Initialize a map 'FREQMAP' to store the frequency of every element.
- Initialize a variable 'MAXFREQ' to store the maximum frequency.
- Initialize a variable 'MINLEN' to store the length of the smallest result subarray.
- Initialize a variable 'STARTIDX' to store starting index of the smallest result subarray.
- Iterate 'I' in 0 to 'N':
    - Set 'X' as 'ARR[I]'.
    - If the current element is not present in 'FREQMAP':
        - Put ('X', 'I') key-value pair in 'LEFT'.
        - Put ('X', 1) key-value pair in 'FREQMAP'.
    - Otherwise, increase the frequency of the current element in 'FREQMAP'.
    - If frequency of current element is greater than 'MAXFREQ':
        - Set 'MAXFREQ' as 'FREQMAP.GET(X)'.
        - Set 'MINLEN' as ('I' - 'LEFT.GET(X)' + 1).
        - Set 'STARTIDX' as 'LEFT.GET(X)'
    - If frequency of current element is equal to 'MAXFREQ' and ('I' - 'LEFT.GET(X)' + 1) is less than 'MINLENGTH':
        - Set 'MINLEN' as ('I' - 'LEFT.GET(X)' + 1).
        - Set 'STARTIDX' as 'LEFT.GET(X)'.
- Initialize an integer array 'ANS' to store the output.
- Initialize a variable 'IDX' to iterate through 'ANS'.

- Iterate 'I' in 'STARTIDX' to 'STARTIDX' + 'MINLEN':
    - Set 'ANS[IDX]' as 'ARR[I]' and increment 'IDX' by 1.
- Return 'ANS'.