

Brute Force

↓
Better

↓
Optimal

Brute Force

generating all the subarrays

3 3 3 1 2 1 1 2 3 3 4



→ Time complexity:

$$O(N \times N) = O(N^2)$$

space complexity: $O(3)$

maxlen = 0

for i = 0 to arr.size() → N

set <int> st;

for (j = i; j < arr.size(); j++) { → N

st.insert(arr[j])

if (st.size() <= 2) {

maxlen = max(maxlen, j - i + 1);

}

}

return maxlen;

Better:

function FB (vector<int> arr, int k) {

maxlen = 0, l = 0, r = 0;

map<int, int> mp;

while (r < arr.size()) {

mp[arr[r]]++;

if (mp.size() > k) {

while (mp.size() > k) {

mp[arr[l]]--;

if (mp[arr[l]] == 0) {

mp.erase(arr[l]);

}

l++;

}

if (mp.size() <= k) {

maxlen = max(maxlen, (r - l + 1));

}

r++;

}

return maxlen;