

```

for(int i = 0; i < k; i++){
    while(dq.size() > 0 && arr[dq.getLast()] < arr[i]) dq.removeLast();
    dq.addLast(i);
}

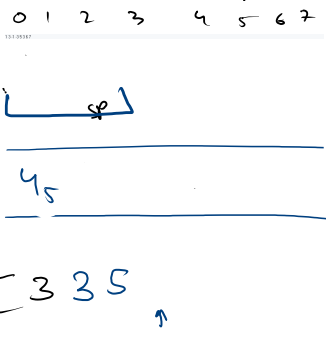
ans[0] = arr[dq.getFirst()];
int sp = 0;
int ep = k;
int window = 1;

while(ep < N){
    //epth person enters the window, so it should be introduced to dq
    while(dq.size() > 0 && arr[dq.getLast()] < arr[ep]) dq.removeLast();
    dq.addLast(ep);

    //it is possible that front of the window is out of the window
    if(dq.getFirst() == sp) dq.removeFirst();

    ans[window] = arr[dq.getFirst()];
    ep++;
    sp++;
    window++;
}
return ans;

```



① consider whole str as  
 ② start sliding window

add ep  
 remove sp  
 ep++  
 sp++  
 $ans[window] = arr[ans[sp]]$

```

int k = 10;
int n = 10;
int arr[] = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10};
int ans[] = new int[n];

//initialization
freq[s.charAt(sp)]++;
if(freq[s.charAt(sp)] == k) charAtLeastTimes++;
//check for sp
while(sp < n){
    freq[s.charAt(sp)]++;
    if(freq[s.charAt(sp)] == k) charAtLeastTimes++;
    sp++;
}

//check for ep
while(ep < n){
    freq[s.charAt(ep)]++;
    if(freq[s.charAt(ep)] == k) charAtLeastTimes++;
    ep++;
}

//check for ans
if(charAtLeastTimes == k) ans[ansIndex] = s.charAt(sp);
else ans[ansIndex] = -1;
ansIndex++;
}
return ans;

```

ans = 5  
 sp = 3  
 ep = 7  
 s = FISH

ans = 5  
 sp = 3  
 ep = 7  
 s = ARIFRISHAB

why  
 F=1  
 I=1  
 R=1  
 S=1  
 H=1

ideal  
 F=1  
 I=1  
 R=1  
 S=1  
 H=1

```

while(ep < s.length()){
    //initialization
    freq[s.charAt(sp)]++;
    if(freq[s.charAt(sp)] == k) uniqueCharacters++;
    //check for sp
    while(sp < s.length()){
        freq[s.charAt(sp)]++;
        if(freq[s.charAt(sp)] == k) uniqueCharacters++;
        sp++;
    }

    //check for ep
    if(uniqueCharacters == k) ans[ansIndex] = s.charAt(sp);
    else ans[ansIndex] = -1;
    ansIndex++;
}
return ans;

```

x=3  
 k=2

sp  
 z y a b a c a b a c p v

uni = 3  
 char = 0

z=0  
 y=1  
 a=1  
 b=1

ans  
 should be used  
 consider as  
 equal

z=1  
 a=2  
 b=3  
 c=1  
 x=1

K=3  
 unique=4  
 unique=3  
 unique=5

ans  
 should be used  
 consider as  
 equal

z=1  
 a=2  
 b=3  
 c=1  
 x=1

S = a b a c a p p d a c s o n u  
 sp

why  
 a=2  
 c=1  
 d=1  
 p=1

ideal  
 a=2  
 c=1  
 d=1  
 p=1

ans[ep]  
 while (used) {  
 consider  
 shrink  
 ep++