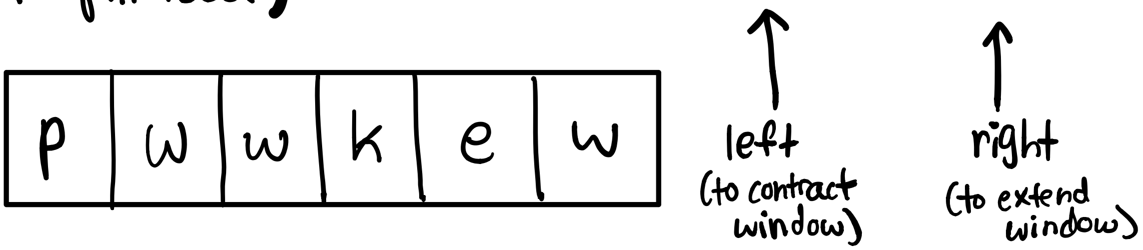
(Not optimized)

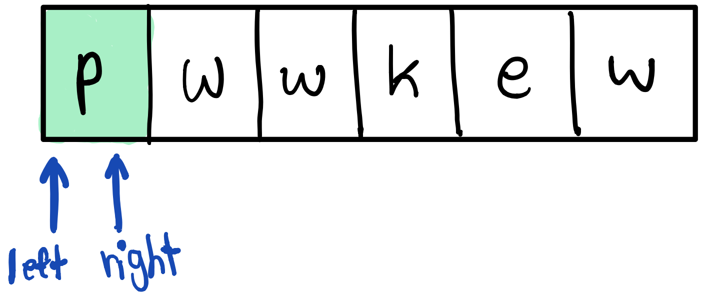


Check for duplicates with:

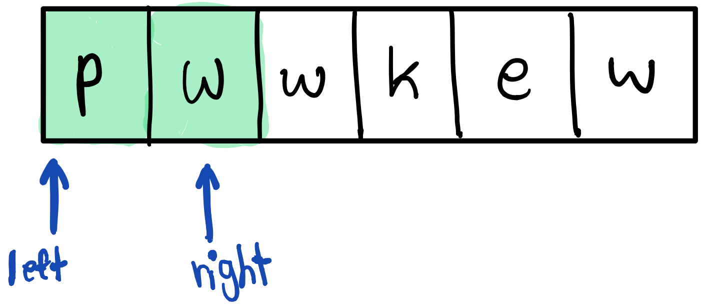
- array
  [arr[ord(char)]+=1]
- hashmap/Counter
  key : count

* at the end of each loop, update max

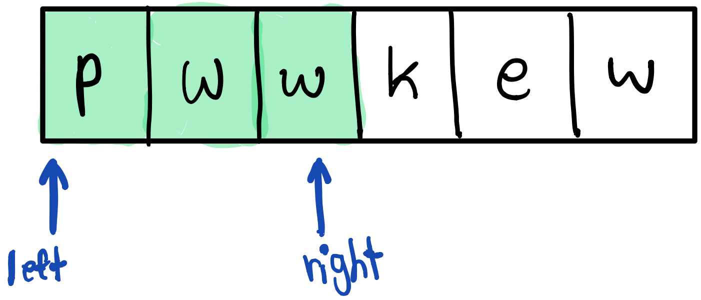
res = max(res, right-left+1)



p:1 no duplicates of p, move right

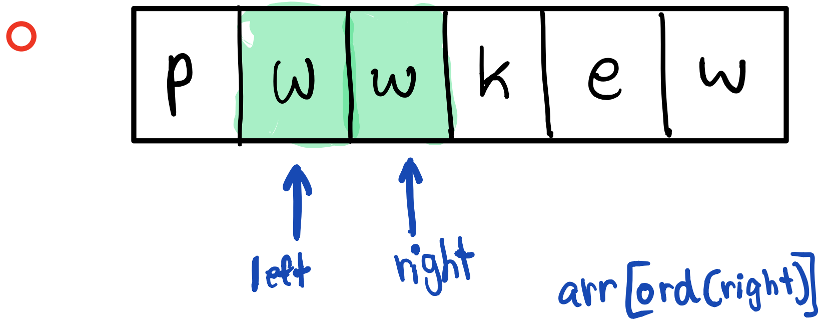


p:1
w:1 no duplicates of w, move right

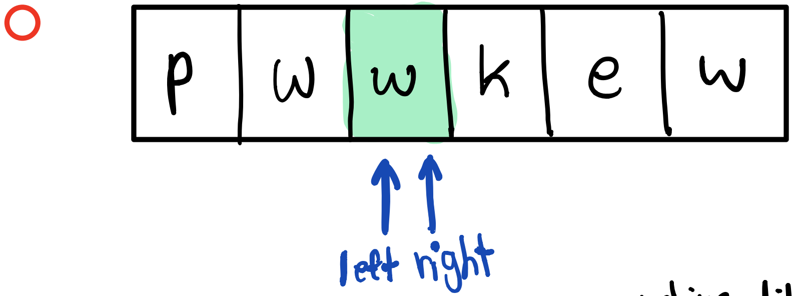


p:1 ! duplicate of w!
w:2

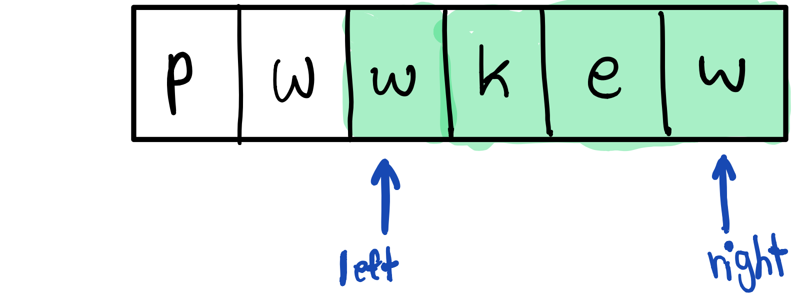
while it's >1, keep moving the left



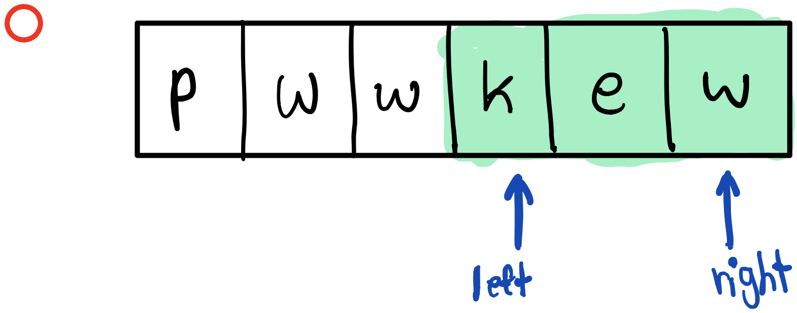
arr[ord(right)]>1   w:2



... keep expanding till end or duplicate



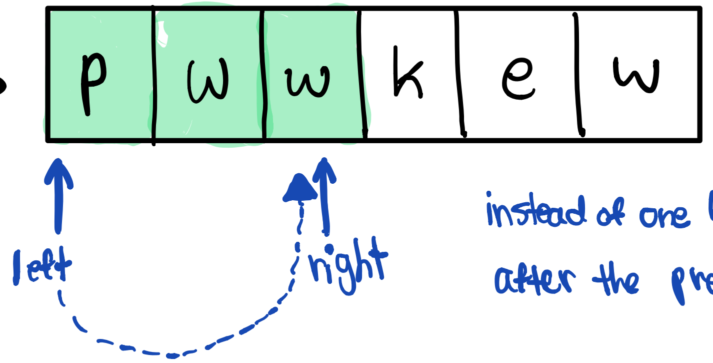
! duplicate, contract until no duplicates



```
while right < len(s):
    move right
    contract while duplicates
    update max
```

[a a a a a] → takes $O(2n) = O(n)$

Space: $O(n)$



instead of one by one, move left to one after the previous w. AKA record previous

$O(n)$ time, $O(m)$ space