04/02/22

DSML Intermediate - DSA

Poll

Strings

Today's Content :

Q1 Toggle case

Q2 Reverse string | Substring , Reverce word by word

Q3 Palindrome, Length of Longest Palindromic Substring

Q4 Sort String

Strings

↳ an array / a list of characters.

↳ Group

$S_1 =$ "abcd"

$S_2 =$ "adcb"

↳ an ordered sequence of characters.

ASCII ⇒ ASCII

American Standard Code for Information Interchange.

How computer stores strings?

↓

Numbers Binary ~ 0/1

"praveen"

	Char	ASCII	Char	ASCII	Char	ASCII
0:	'a'	97	'A'	65	'0'	48
1:	'b'		'B'		'1'	
2:	'c'		'C'			
⋮	⋮		⋮		⋮	
25:	'z'	122	'Z'	90	'9'	57

ord('A') = 65

ord('a') = 97

char ⟶ ASCII

chr(65) = 'A'

chr(97) = 'a'

ASCII ⟶ char

Q1. Given a string S, toggle case of every char

lower → upper

upper → lower

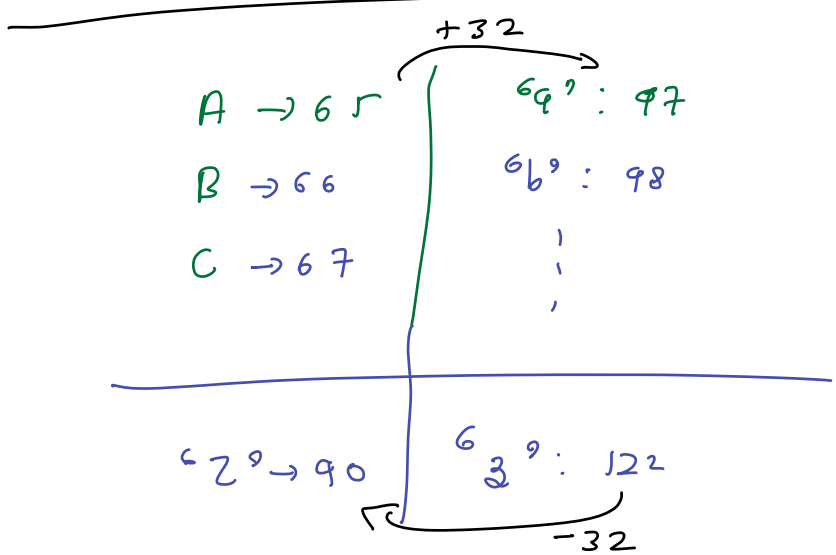
Can use only chr/ord.

I/p: AbCdEfGh

O/p: aBcDeFgH

Capital has smaller ASCII

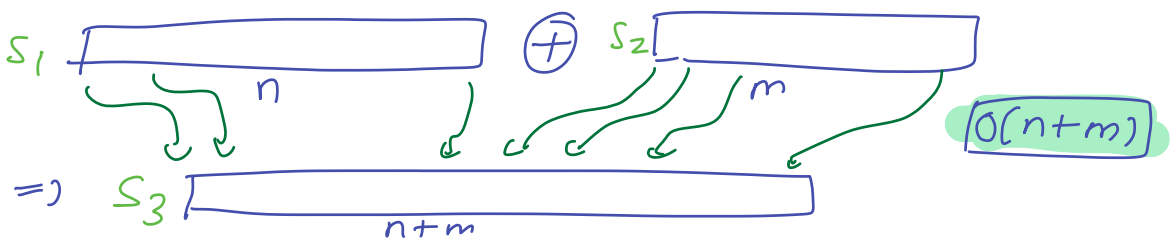
→ 97
→ 65



Upper —(+32)→ Lower

Lower —(-32)→ Upper.

String Concatenation



```
s # input
res = ""
for i in range (0, len(s)):
    if ord(s[i]) >= 65 and ord(s[i]) <= 90:
        new_ascii = ord(s[i]) + 32
        res += chr(new_ascii)
    elif ord(s[i]) >= 97 and ord(s[i]) <= 122:
        new_ascii = ord(s[i]) - 32
        res += chr(new_ascii)
```

n = 6

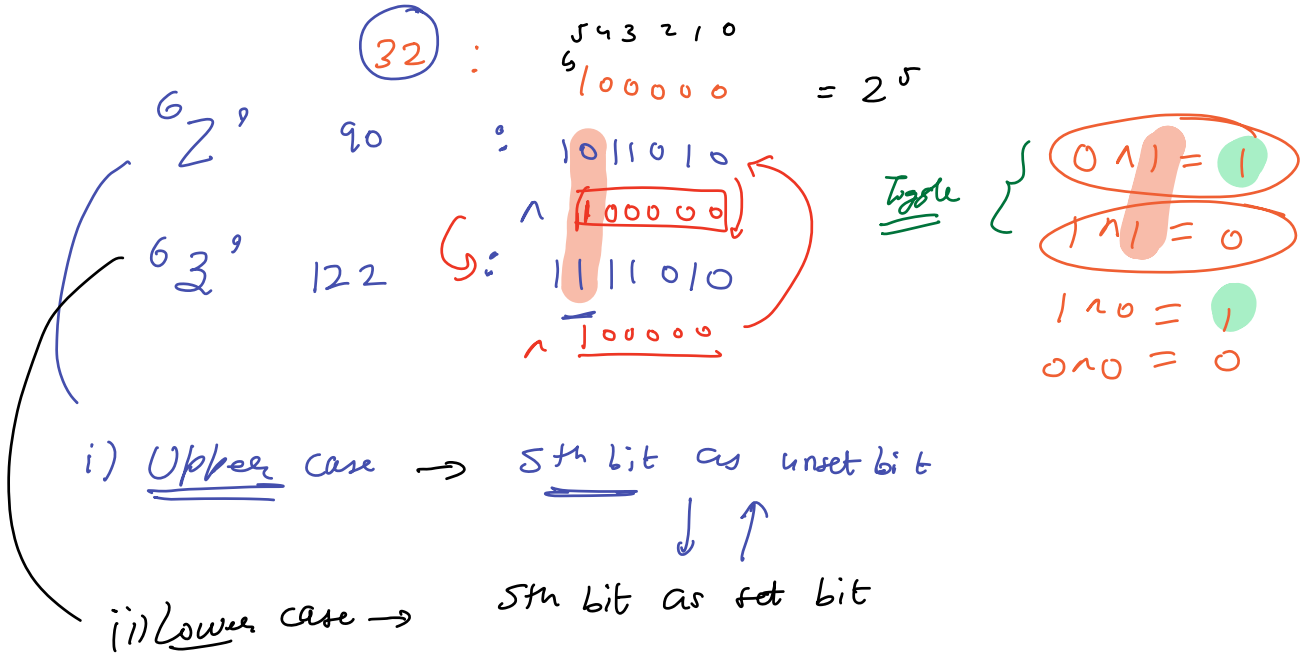
Manthoj

0	1		Concatenation
""	+ m	= "m"	0+1 = 1
"m"	+ "A"	= "mA"	1+1 = 2
"mA"	+ "N"	= "mAN"	2+1 = 3
⋮		⋮	⋮
"mANAO"	+ "J"	= [0,1,2,3,4,5,6]	5+1 = 6

$1 + 2 + 3 + \cdots + N = \frac{N(N+1)}{2} = O(N^2)$

Take Aways

1. Strings are immutable
2. Concatenation is O(N) time
3. str —O(N)→ convert list of chars → solve (?) —O(N)→ join. O(N)



i) Upper case → 5th bit as unset bit

ii) Lower case → 5th bit as set bit

Upper → Lower +32 → set 5th bit

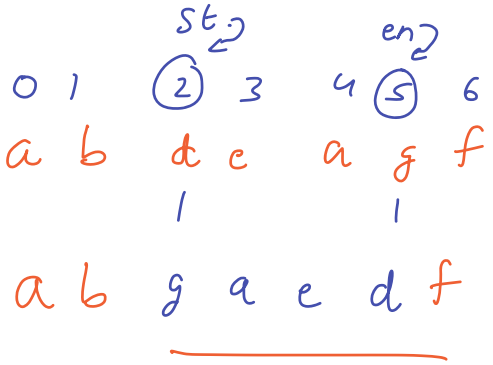
Lower → Lower −32 → Unset 5th bit

} Toggle 5th bit.

[ ]

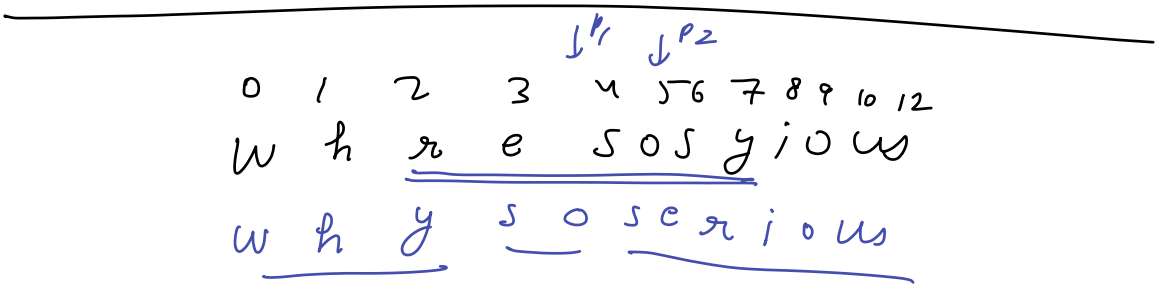
empty string → ''

Q2. Given a string, reverse a part of string.



substring.

- Concept same as subarrays.
- Continuous part of string
- Full string can be substring
- empty string is also ""
- one char is also ""



List — Input is a list of chars not string.

```
def reverse(s, st, en):
    p1 = st
    p2 = en
    while(p1 < p2):
        swap(s[p1], s[p2])
        p1 += 1
        p2 -= 1
    return ''.join(s)
```

p1 = 0

p2 = N-1

Entire string.

Time = O(N)

Space = O(N) → O(1) excluding ans join.

Q3 Given a char array, reverse it word by word.

Google
Amazon
Microsoft

S = "love hate data structures"

→ No space at beg/end
→ Only 1 space b/w consecutive words
→ No inbuilt fn
→ Do it in O(1) space excluding final join
→ O(N) time.

i) I/P = ['e', 'o', 'v', 'e', ' ', 'h', 'a', 't', 'e', ' ', 'd', 'a', 't', 'a', ' ', 's', 't', 'r', 'u', 'c', 't', 'u', 'r', 'e', 's']

O/P = "structures data hate love" ← join

ii) I/P: "danger"

O/P: "danger"

I/P: "Mailman brings letters"

O/P: "letters brings Mailman"

Brute: Split Input: ["Mailman", "brings", "letters"]

↓ reverse array of strings

["letters", "brings", "Mailman"]

join ← (yes)

s : "Mailman brings letters"

Step 1 : Rev entire string

rev(s) : "sretteL sgnirb namliaM"

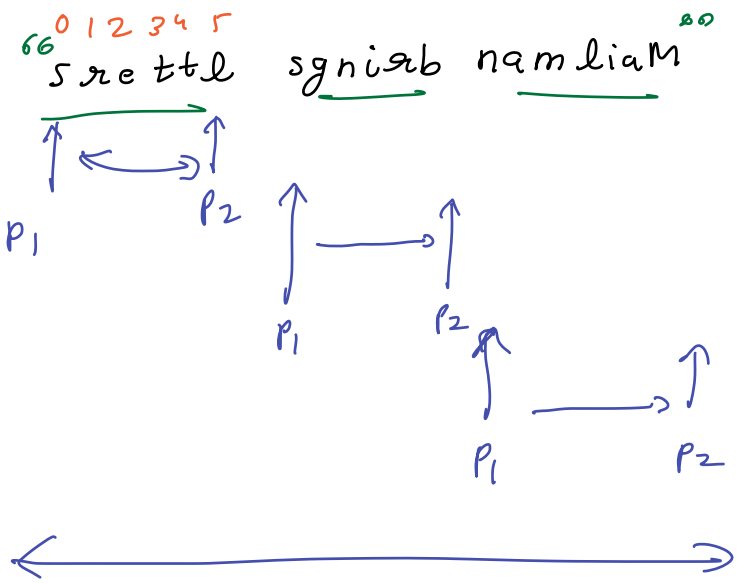
Step-2 : Identify & rev each word (substring) individually

res = letters brings Mailman

Time = O(N)

Space = O(1) excluding any joins.

Step-2 :



$p_1 = 0$

$p_2 = 0$

$(p_1 < N)$ or $(p_2 < N)$

```
while (          ) :
    # identify the word.
    while(p2 < N    and    s[p2] != ' '):
        p2 ++
    # (p1, p2-1)
    reverse( s, p1, p2-1)
    # update p1 & p2
    p1 = p2 + 1
    p2 = p1
```

In total $O(N)$

p_2: $0 \rightarrow N-1$

In total $O(N)$ $N/2$

(0, 1)
(3, 4)
(6, 7)
(9, 11)

---

a b a b a b a b c

0 1 2 3 4 5 6 7 8 9 10 11

1 1 1 1

p_1 p_2

Outer loop

4

#words $K + N + N$ $K << N$

$= O(N)$

Q 4 Given a string, check if it is a palindrome ?

→ mom
→ dad
lol
nitin
naman
wow
noon
refer
madam

malayalam
111
nayan

Approach-1

a ✓ ↕ len = 1 is a palindrome ✓

Rev & Check Equal

s = ab ✗

rev(s) = ba

s = abc ✗

rev(s) = cba

s = aba ✓

rev(s) = aba.

Approach-2

a b c c b a
↑ ↑
p2 p1

c b a d a b c
↑
p1
↑
p2

1)

```
p1 = 0
p2 = N-1
while (p1 < p2):
    if (s[p1] != s[p2])
        return False
    p1 += 1
    p2 -= 1
return True
```

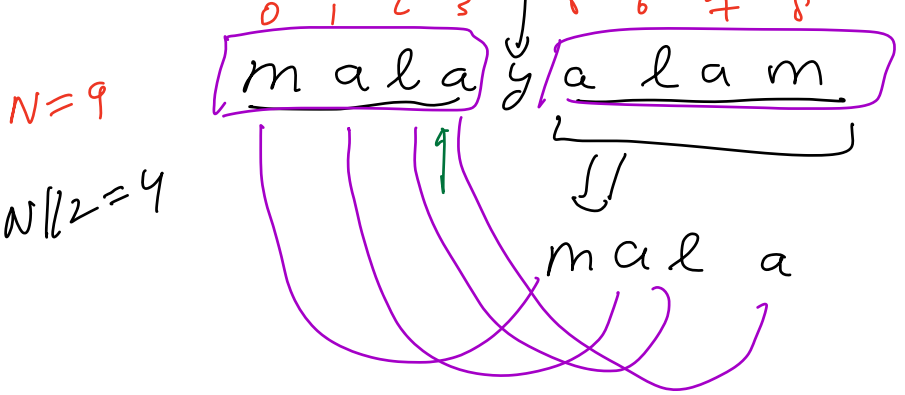
N/2 => O(N) Time.

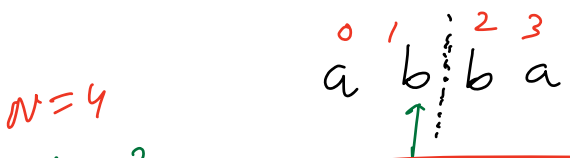
Space = O(

Approach - 3

centre

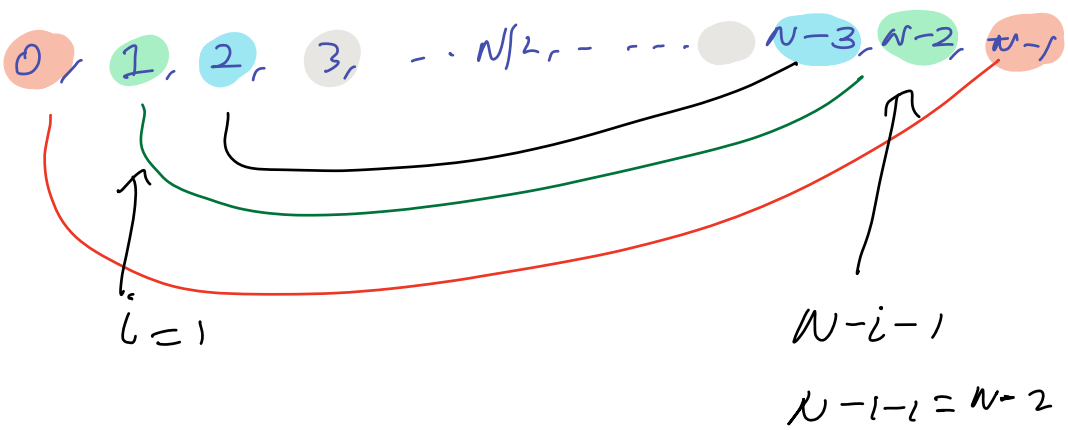
From centre, left & right half are mirror images.






```
for i in range(0, N//2):
    if s[i] != s[N-i-1]
        return False
return True
```

O(N/2) = O(N)



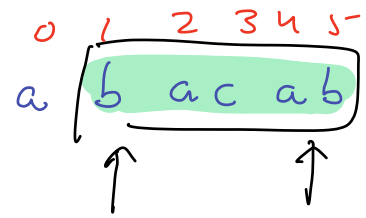
Q 5. Length of longest palindromic substring.

= palindrome

has longest possible length.

odd / even.

Eg. 1



Eg 2 a b c d e am = 1

Brute →

" " ↑ len(" ") = 0

a b c

a b c

a b b c

a b c

3 + 2 + 1

n + (n-1) + (n-2) + ... + 1

$= \frac{n(n+1)}{2}$

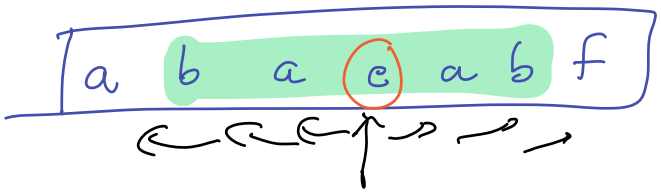
" " Space

$\frac{3 * 4}{2} = 6$

$O(N^2) * O(N) = O(N^3)$

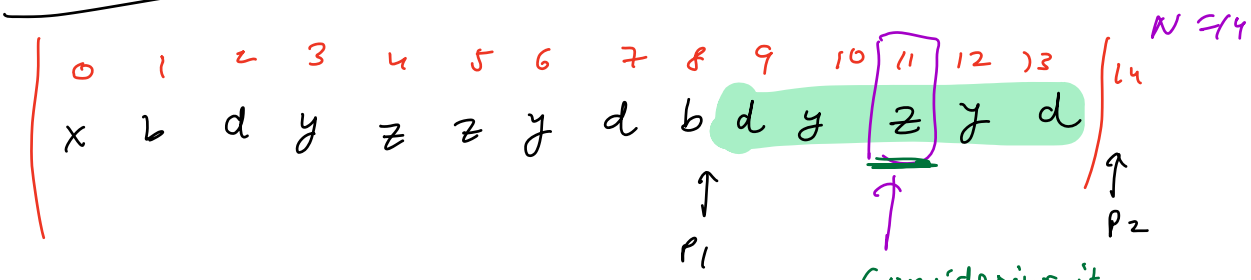
∀ substr — check pali

take max len



(1) If we know the center of Longest Palindromic Substring => we are done.

Do we know it? NO



→ max len of pali given centre

def expand(s, p1, p2):

while(p1 >= 0 and p2 < N):

p1 -= 1

p2 += 1

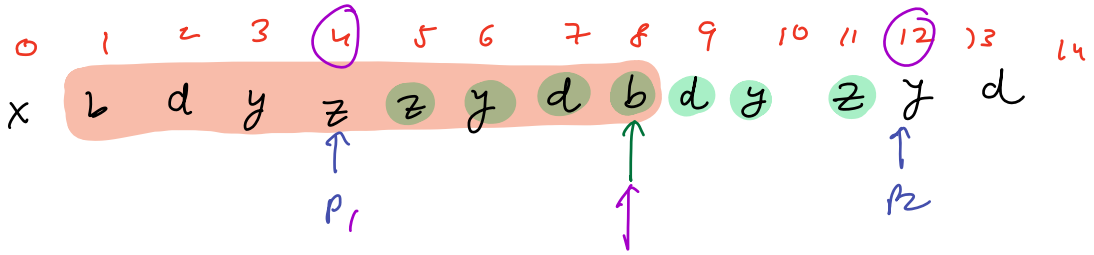
p1 = 8, p2 = 14 → 14 - 8 - 1 = 6 - 1 = 5

return p2 - p1 - 1 # len

Find center

Take every elt as centre & expand around it.

Take max len. ⇒ Max odd len palindrome




```
def expand(s, p1, p2):
    while(p1>=0 and p2<N):
        p1 -= 1
        p2 += 1
    # p1 = 8, p2 = 14
    return p2-p1-1   # len
```

O(N)

max = 1 → max = 7

Odd

```
maxi = 1
for i in range(0, N):
    maxi = max(maxi, expand(s, i, i))
```

Even

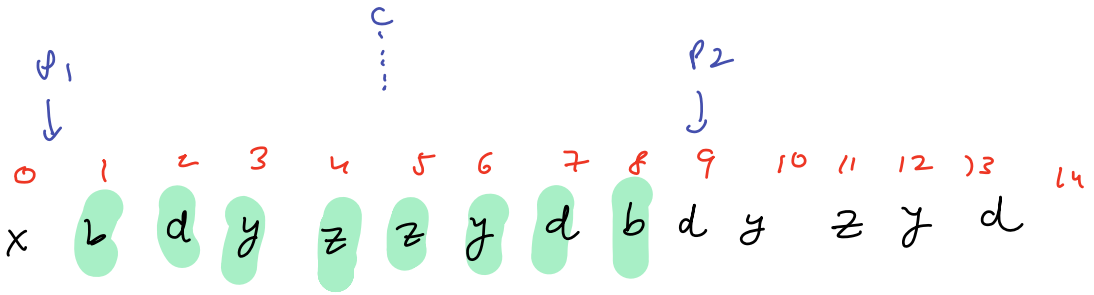
```
maxi = max(maxi, expand(s, i, i+1))
```

Time = O(N*N) = O(N^2)

Space = O(1)

Even Length Max Palindrome

Max = 8



$p_2 - p_1 - 1 = 9 - 0 - 1 = 8$

$N^3 \rightarrow N^2 \rightarrow O(N)$

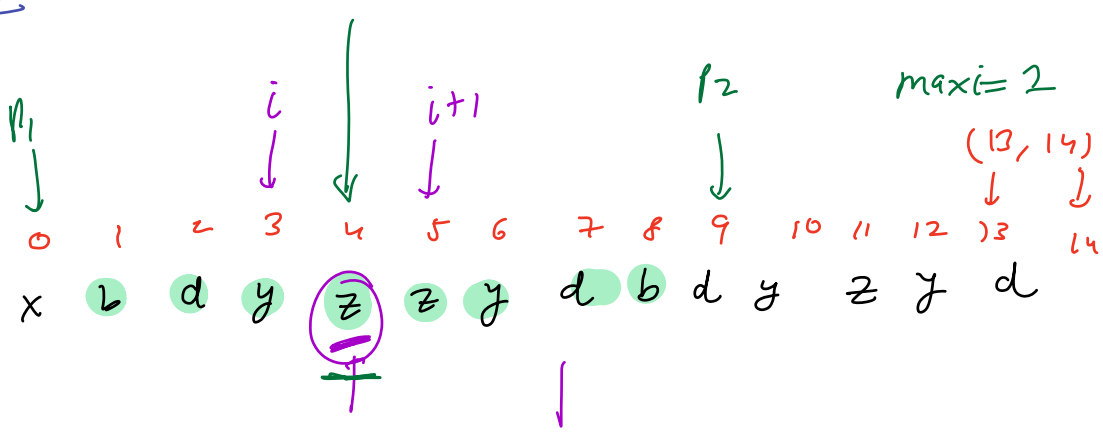
(Manacher's Algo)

Advanced Batch

DP: $O(N^2)$

SC: $O(N^2)$

Doubts

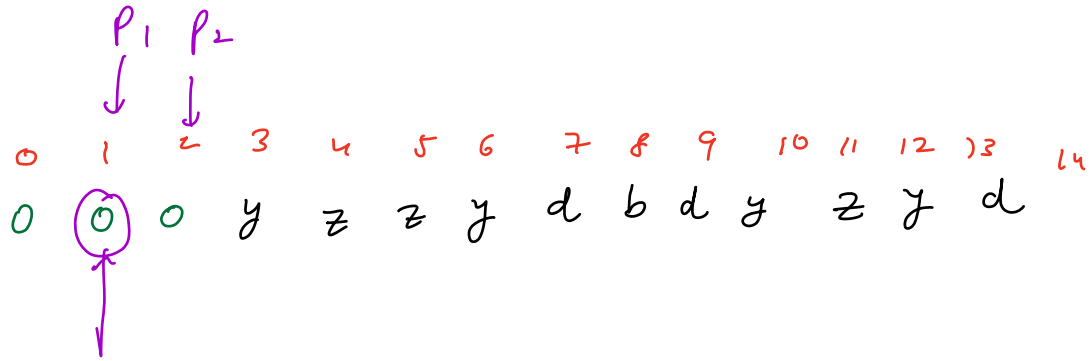


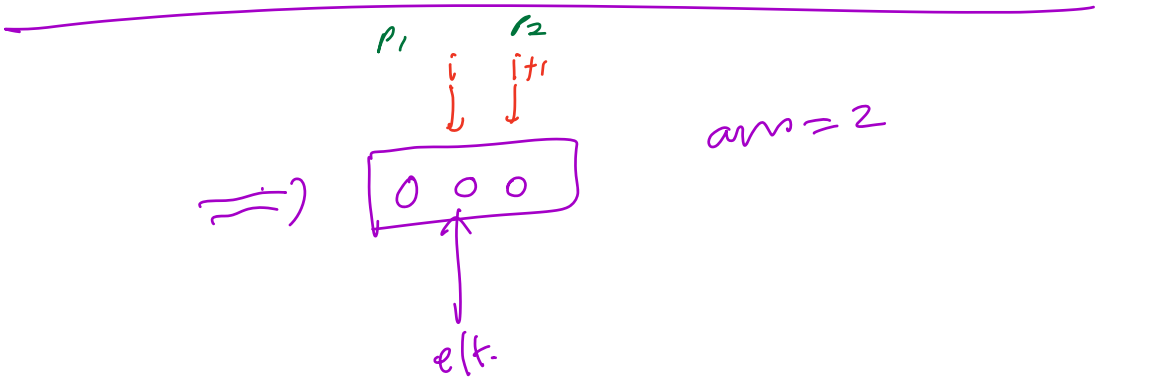

```
def expand( s, p1, p2):
    if ( p1>=0 and p2<N and s[p1] ! s[p2]){ p1--;}
    while( p1>=0 and p2< N):
        p1 -= 1
        p2 += 1
# p1 = 8, p2 = 14
    return p2-p1-1   # len
```

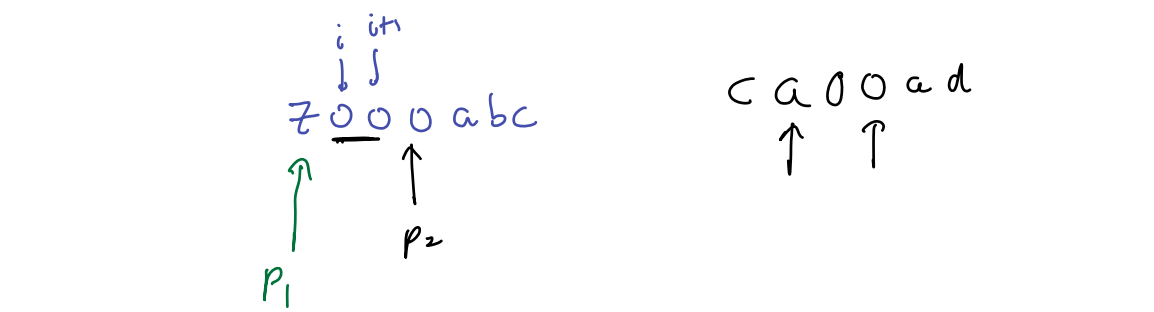
main

```
maxi = 1
for i in range(0,N):
    maxi = max(maxi, expand(s, i, i))      odd
    maxi = max(maxi, expand(s, i, i+1))
```

0, 1
1, 2
2, 3
.
.
N-1, N


p1
p2
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14
0 0 0 y z z y d b d y z y d


p1
p2
i
i+1
0 0 0
elt.
ans = 2


i
i+1
z 0 0 0 a b c
p1
p2
c a 0 0 a d

=> Doubts Pending

Hashing 1 & 2

W F

① * Remedial session → Mon ← Next 7th Feb

Key concepts

+ Problems

② Problem solving session => Extra on Next Sun

↳ Assignment + HW (Stuck on observations etc.)

Slack → Poll

Class → Let me know.

Backlog