Longest Palindromic Substring

→ if odd lengths kaa answer find out karna hai.

10

0 1 2 3 4 5
a b a a b b
↑ ↑
i i

first for a
↳ j step left mai & j step right mai move karke check karenge ki hai ki nhi Palindrom

for a → max palindrom of odd length is 1 → a itself

at b(1)
b itself palindrom so
j = i-1 & k = i+1
& mapp S[j] == S[k]
if yes palindrom increase.

for odd length we can findout this way

```
for(i=0 → n)
    int k=0
    while(i-k>=0 & i+k<n & S[i-k] == S[i+k]) {
        k++;
    }
    k--;
    if((i+k)-(i-k)+1 > maxlen){
        s = i-k;
        maxlen = (i+k)-(i-k)+1;
    }
}
```

Today's Priorities

Birthday / Anniversary

Harsh words break no bones of yours but they do break hearts.

JUNE						2013
M	T	W	T	F	S	S
					1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30

even length substring
↳ It is center around two center character

→ same as odd one bas farak itna hi hai
isme hi isme mai i & j do se lung

abaa bb

& inke around Palindrome findout karunga konsa hai.

is ka TC → $O(n^2)$

Today's Priorities

29 June
Saturday
180-185 Week-26

JUNE 2013
M T W T F S S
1 2
3 4 5 6 7 8 9
10 11 12 13 14 15 16
17 18 19 20 21 22 23
24 25 26 27 28 29 30

Longest Palindromic Substring
MANACHEER'S Algorithms

→ In manacheer's Algorithms

→ Step 1: Jo humari String hai usmai start & end mai special characters add kardo
→ babad
→ newString ⇒ #b#a#b#a#d#
→ taki hume har ek string ke around odd type ka hi Palindrome mile baad mai e piche odd wali condition laga kar vector of p Fill kardo

	0	1	2	3	4	5	6	7	8	9	10	11	12
t	#	b	#	a	#								
p	0												

	0	1	2	3	4	5	6	7	8	9	10
t	#	b	#	a	#	b	#	a	#	d	#
p	0	1	0	1	0	3	0	1	0	1	0

ab jo sbka max hoga p ka wahi apna max length hogi

a#b#a remove # & concatinate again.

Today's Priorities

Birthday / Anniversary

Don't ask for a light load, but rather ask for a strong back.

code decode

T = "^ # a # b # a # b # a # d # $"
n = 13

P =

0	1	2	3	4	5	6	7	8	9	10	11	12
.		1		3								
		↑										

C=0, R=0
i=1
mirror = 2*C - i
= -1
if (i < R) ✓
P[i] = min(R-i, P[-1]); X
while (T[i+P[i]+1] == T[i-P[i]-1]) X
if (i + P[i] > R)
1 + 0 > R)
C = 1 C=1
R = 1+0 R=1

i=2
mirror = 4 - 2 = 2
if (2 < 1) X
while (T[2+0+1] == T[2-0-1])
T[3] == T[1] ✓
P[i]++
if (1+1 > 1) ✓
C=2 C=2
R=2+1 R=3

C=2 R=3, i=3
mirror = 4 - 3 = 1
if (R > i) X
while (T[3+P[3]+1] == P[3-P[3]-1]
T[4] == T[2] X

if (i + P[i] > R)
3 + 0 > 3 X

i=4, C=2 R=3
mirror = 4 - 3 = 1
if (3 > 4) X

while (T[4+P[4]+1] == T[4-P[4]-1])
P[i]++
T[4+1+1] == T[4-1-1]
T[6] == T[2] ✓
P[i]++
T[4+2+1] == T[4-2-1]
T[7] == T[1] ✓

Always wear the costume of humility and you will receive the love and co-operation of others.

NOTES

till similarly 2.

we will get at the end of this

0	1	2	3	4	5	6	7	8	9	10	11	12

P = [0, 0, 1, 0, 3, 1, 3, 1, 0, 1, 0, 1, 0]

max_len = 0
centres = 0

max_len = 3
centres_index = 1

start = (4 - 3) / 2.