abaaabca

aba

$O(n^2)$ using dp

$dp[i,j] = dp[i+1,j-1]$ and $s[i]==s[j]$

$S = abba$ $\quad N = 4$

$\Downarrow$

$T = \#a\#b\#b\#a\#$ $\quad N' = N+N+1 = 2N+1$

$\Downarrow$

Apply Manacher's algorithm

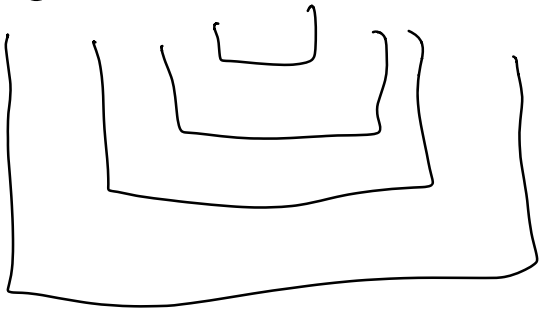
(This can only find odd lengths palindrome)

Manacher's Algorithm

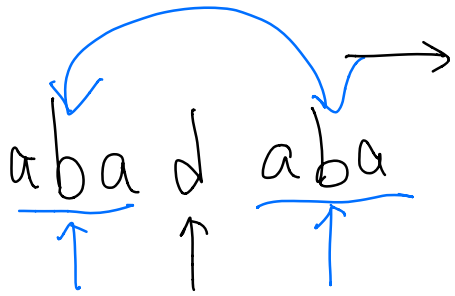
Intuition:

a palindrome has a mirror property

Ex. e d c b a b c d e



Ex.



You can see that if b can expand to aba its mirror b can ALSO do it!

Given a string S

and $T = transform(S, \#) = a_0 a_1 a_2 \cdots a_{n-1}$

Define $P[\,]$ where $len(P) = len(T)$

and $P[i]$ means $a_{i-P[i]} \sim a_{i+P[i]}$ is a palindrome

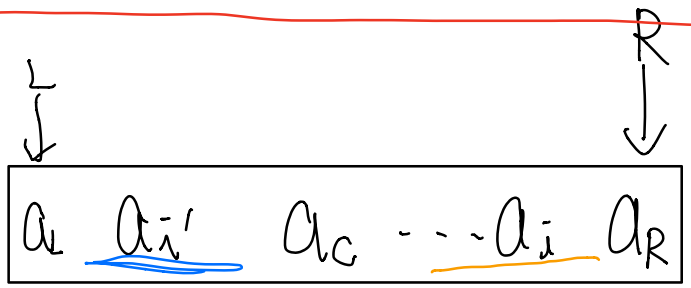
Let's assume we want to compute

$P[i]$ ($P[j]$ where $j < i$ is computed)

Define $i' = mirror(i, C) = 2C - i$

$(\Rightarrow i' - C = C - i \Rightarrow i' = 2C - i)$

Case 1: ⟶ does not touch a_L



Observe

$a_{i'-P[i']} \cdots a_{i'} \cdots a_{i'+P[i']}$

Consider k, $0 \le k \le P[i']$

$$a_{i+k} = a_{\text{mirror}(i+k, c)}$$

$$= a_{2c-(i+k)}$$

$$= a_{(2c-i)-k}$$

$$= a_{i'-k}$$

$$a_{i-k} = a_{\text{mirror}(i-k, c)}$$

$$= a_{2c-(i-k)}$$

$$= a_{(2c-i)+k}$$

$$= a_{i'+k}$$

$\Rightarrow a_{i+k} = \underline{a_{i'-k}}$, $a_{i-k} = \underline{a_{i'+k}}$

Because $a_{i'-k} = a_{i'+k}$

$\Rightarrow a_{i+k} = a_{i-k}$

And we already know $a_{i'+P[i]+1} \neq a_{i'-P[i]-1}$

How about $a_{i+P[i]+1}$ and $a_{i-P[i]-1}$

$$a_{i+P[i]+1} = a_{\text{mirror}(i+P[i]+1,\, c)}$$
$$= a_{2c-(i+P[i]+1)}$$
$$= a_{(2c-i)-P[i]-1}$$
$$= a_{i'-P[i]-1}$$

$a_{i-P[i]-1} \overset{\text{using the same technique}}{=\!=} a_{i'+P[i]+1}$

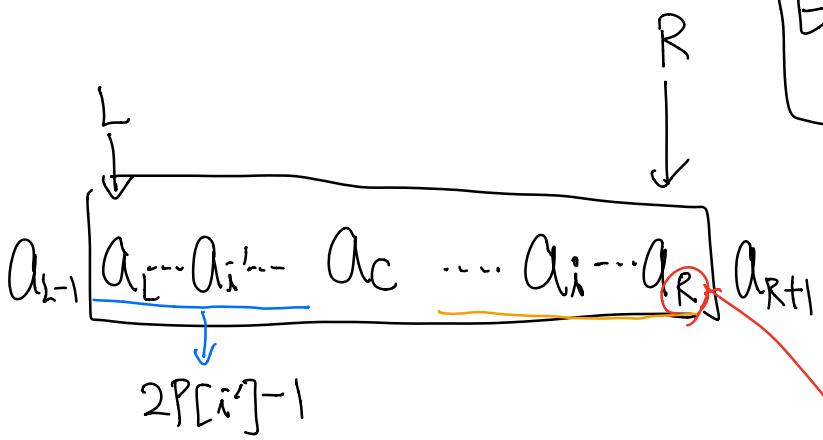
$\Rightarrow a_{i+P[i]+1} \neq a_{i-P[i]-1}$

So $\boxed{P[i] = P[i']}$

$(i' = \text{mirror}(i, C))$

Case 2: ——— touches a_L

Ex. c|babtbab|cab



Can we say $P[i] = P[i']$?

By the same reasoning as Case 1,

a_L --- $a_{i'}$ ---- $a_{i'+P[i']}$

$a_{i'-P[i']}$ ||

$a_{i+P[i']}$ --- a_i ---- $a_{i-P[i']}$

Note: $a_{i+P[i']} = a_L$

and $a_L = a_{mirror(L,C)} = a_R$

$\Rightarrow a_{i+P[i']} = a_R$

Is $a_{i+P[i']+1} = a_{i-P[i']-1}$?

Observe $a_{L-1} = a_{i'-P[i']-1} \neq a_{i'+P[i']+1}$

However, knowing this property does not help us determine whether $a_{R+1} = a_{i+P[i']+1} \stackrel{?}{=} a_{i-P[i']-1}$

$\Rightarrow$ We must still expand it!

Case 3: ——— expand beyond a_L

Ex. abababacb

cannot be a!

$a_L \cdots a_{i'} \cdots \; a_c \quad a_i \; a_R$

Can we determine $P[i]$ by $P[i']$?

Let us assume $P[i]=P[i']$

$a_{i'-P[i']} \; a_L \; - - - \; a_{i'} \; - - - \; a_{i'+P[i']}$

$\downarrow \qquad \downarrow \qquad \downarrow \qquad \downarrow$

$a_{i+P[i']} \; a_R \; - - - \; a_i \; - - - \; a_{i-P[i']}$

By $a_{i'-k} = a_{i+k}$ (where $i'-k < L$)

If we consider $a_{\text{mirror}(i'-k,\, c)}$

$= a_{2c-(i'-k)}$

$= a_{(2c-i')+k} = a_{i+k}$

That means, L, R can be expanded even more!

Contradicts that R is the farest we can expand until know!

Thus, $P[i] = R - i$

Because it cannot be expanded further

Case 4: $[L, R]$ does not cover i

We have no information to use.

So we just expand a_i left and right.

Time Complexity:

Because we will at most visit each element twice.

One is by expanding.

One is by using its mirror information.

$\Rightarrow O(n)$

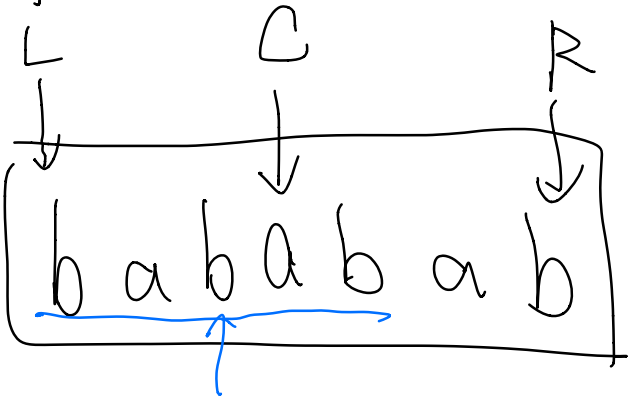
Or you can say because R will never go backward and i will only go right too.

This leads us an $O(n)$ solution

FAQ:

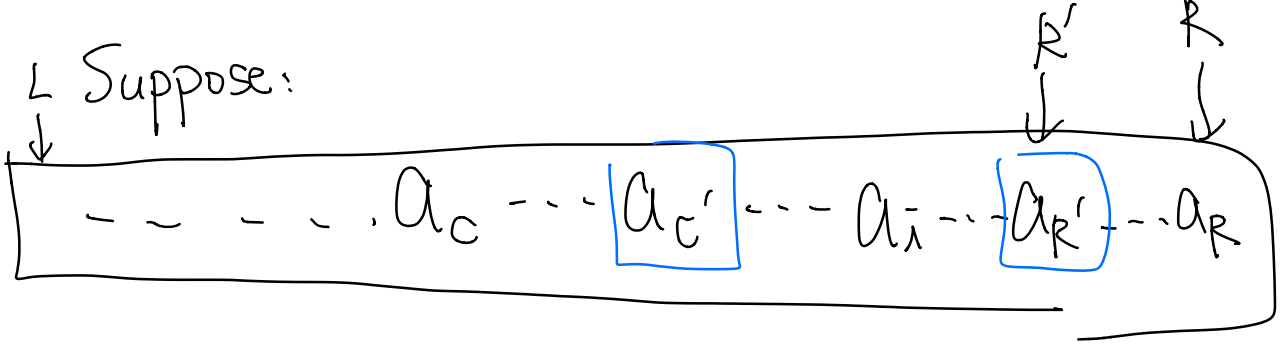
Q: Can $\longrightarrow$ cover a_c

A: Yes, but it does not affect our derivation



Q: Why choose the rightest R?

A: Because choosing an $R' < R$ will lead us no benefit.



Using R' will make us re-expand if we encounter case 2

So why not choose R?