

Count Palindromic Substrings

a d b c c b a

st ↓

→ et

	0	1	2	3	4	5	6
	a	d	b	c	c	b	a
a 0	T	F	F	F	F	F	F
d 1	X	T	F	F	F	F	F
b 2	X	X	T	F	F	T	F
c 3	X	X	X	T	T	F	F
c 4	X	X	X	X	T	F	F
b 5	X	X	X	X	X	T	F
a 6	X	X	X	X	X	X	T

i j → $ch(i) == ch(j)$
 if $pal(i+1, j-1)$

count : total no. +

longest : last true
 (diagonal)

Longest Palindromic Subsequences

str: a b c k b

brute force : 2^n

all palindromic subseq.

a

b b

b c b

b

c

b

↳ dp5

longest pal
subseq.

ans: 3

$$dp(s) = dp(c1 \text{ m } c2) = S(c1 \text{ m } c2)$$

c1 -> left ext

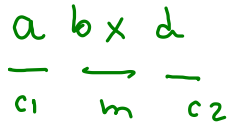
c2 -> right ext

$$= - s(m) - \quad (1)$$

$$= s(m) c2 \quad (2)$$

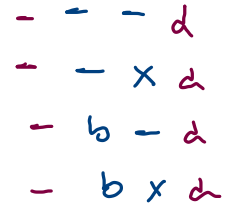
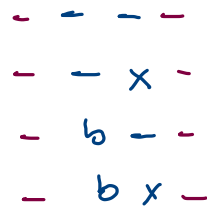
$$c1 \ s(m) - \quad (3)$$

$$c1 \ s(m) \ c2 \quad (4)$$



$$\boxed{- s(m) -}$$

$$\boxed{- s(m) c2}$$



$$c1 = c2$$

$$(4)$$

$$c1 \neq c2$$

$$(1), (2), (3)$$

$$dp(s) = dp(m) + 2$$

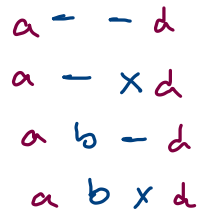
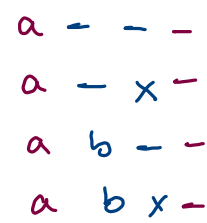
$$dp(s) = \max((1), (2), (3))$$

$$= \max(dp(c1 \text{ m}), dp(m \text{ c2}))$$



$$\boxed{c1 \ s(m) -}$$

$$\boxed{c1 \ s(m) c2}$$



$$z = s(m) - \quad (1)$$

$$- s(m) c_2 \quad (2)$$

$$c_1 s(m) - \quad (3)$$

$$c_1 s(m) c_2 \quad (4)$$

$$\underline{c_1 z = c_2}$$

$$a \underline{b k g e k a} a$$

$$3 \rightarrow \underline{a} \underline{k g k a}$$

$$a \underline{k g k} a$$

$$\text{dps}(\underbrace{c_1 m c_2}_{\text{Str}})$$

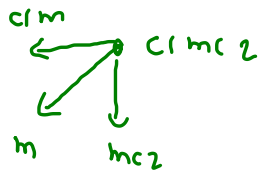
$$\left\{ \begin{array}{ll} \text{dps}(m) + 2 & c_1 == c_2 \\ \max \left(\begin{array}{l} \text{dps}(c_1 m), \\ \text{dps}(m c_2) \end{array} \right) & c_1 != c_2 \end{array} \right.$$

$\text{dps}(m) \rightarrow i+1, j-1$

$\text{dps}(c_1 m) \rightarrow i, j-1$

$\text{dps}(m c_2) \rightarrow i+1, j$

string: a b k c c b c



	a_0	b_1	c_2	c_3	c_4	b_5	c_6
a_0	1	1	1	1	2	4	4
b_1	X	1	1	1	2	4	4
c_2	X	X	1	1	2	2	3
c_3	X	X	X	1	2	2	3
c_4	X	X	X	X	1	1	3
b_5	X	X	X	X	X	1	2
c_6	X	X	X	X	X	X	1

$\text{dp}(i, j) \rightarrow$ longest palindromic subseq of string from i to j .

Count Palindromic Subsequences

abc**k**b

a b b b c b
b b k b
c
k
b

Count = 8

Count Palindromic Subsequences

$$\begin{aligned}
 cps(str) &= cps(c_1 m c_2) = S(c_1 m c_2) \\
 &= - S(m) - \quad \textcircled{1} \\
 &\quad - S(m) c_2 \quad \textcircled{2} \\
 &\quad c_1 S(m) - \quad \textcircled{3} \\
 &\quad c_1 S(m) c_2 \quad \textcircled{4}
 \end{aligned}$$

$$\boxed{\textcircled{4} = \textcircled{1} + 1}$$

↙
c₁c₂

$$c_1 == c_2$$

$$\textcircled{1} + \textcircled{2} + \textcircled{3} + \textcircled{4}$$

$$cps(str) = \textcircled{1} + \textcircled{2} + \textcircled{3} + \textcircled{4}$$

$$= \textcircled{1} + \textcircled{2} + \textcircled{3} + \textcircled{1} + 1$$

$$\swarrow \quad \quad \quad \searrow$$

$$cps(mc_2) + cps(c_1m) + 1$$

$$c_1 \neq c_2$$

$$\textcircled{1} + \textcircled{2} + \textcircled{3}$$

$$cps(str) = \textcircled{1} + \textcircled{2} + \textcircled{3} + \textcircled{1} - \textcircled{1}$$

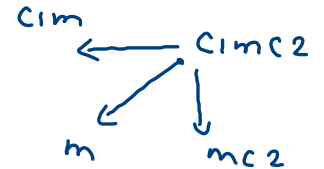
$$= \underbrace{\textcircled{1} + \textcircled{2}}_{\downarrow} + \underbrace{\textcircled{3} + \textcircled{1}}_{\downarrow} - \underbrace{\textcircled{1}}_{\downarrow}$$

$$= cps(mc_2) + cps(c_1m) - cps(m)$$

$$+ cps(c_1m) - cps(m)$$

$$\text{cps}(\text{str}) = \text{cps}(c_1 m c_2) = \begin{cases} \text{cps}(c_1 m) + \text{cps}(m c_2) + 1 & c_1 = c_2 \\ \text{cps}(c_1 m) + \text{cps}(m c_2) - \text{cps}(m) & c_1 \neq c_2 \end{cases}$$

	a_0	b_1	k_2	c_3	c_4	b_5	c_6
a_0	1	2	3	4	6	12	18
b_1	X	1	2	3	5	11	17
k_2	X	X	1	2	4	5	11
c_3	X	X	X	1	3	4	10
c_4	X	X	X	X	1	2	5
b_5	X	X	X	X	X	1	2
c_6	X	X	X	X	X	X	1



i, j \rightarrow count of palindromic
subseq of string
from i to j.

Longest Common Subsequence

abcd
aebd

a b c d

a e b d

Common SS : a b d ab ad bd

abcd

↳ LCS

$$lcs(s_1, s_2) = lcs(c_1 r_1, c_2 r_2)$$

$$= s(c_1 r_1) \times s(c_2 r_2)$$

$$= \begin{bmatrix} -s(r_1) \\ c_1 s(r_1) \end{bmatrix} \times \begin{bmatrix} -s(r_2) \\ c_2 s(r_2) \end{bmatrix}$$

$$= -s(r_1) \times -s(r_2) \quad (1)$$

$$-s(r_1) \times c_2 s(r_2) \quad (2)$$

$$c_1 s(r_1) \times -s(r_2) \quad (3)$$

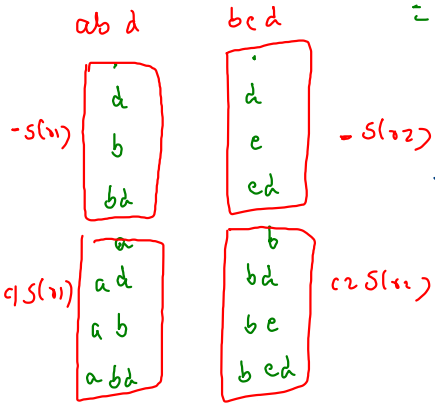
$$c_1 s(r_1) \times c_2 s(r_2) \quad (4)$$

$c_1 \rightarrow$ starting char of s_1

$r_1 \rightarrow$ remaining str of s_1

$c_2 \rightarrow$ starting char of s_2

$r_2 \rightarrow$ remaining str of s_2



$c_1 = c_2$

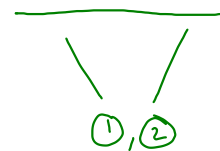
(4)

$$lcs(s_1, s_2) = lcs(r_1, r_2) + 1$$

$c_1 \neq c_2$

(1), (2), (3)

$$lcs(s_1, s_2) = \max(lcs(c_1 r_1, r_2), lcs(r_1, c_2 r_2))$$



$$dcs(s_1, s_2) = dcs(c_1 r_1, c_2 r_2) =$$

$$dcs(r_1, r_2) + 1$$

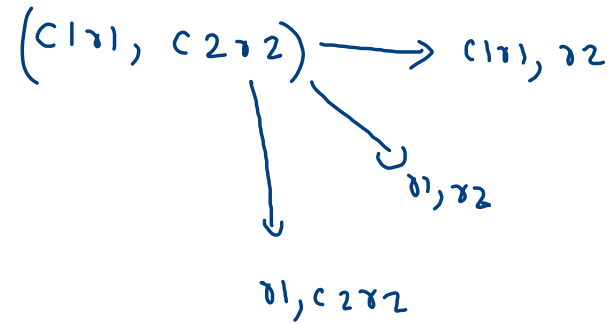
$$c_1 = c_2$$

$$\max(dcs(c_1 r_1, r_2), dcs(r_1, c_2 r_2))$$

$$c_1 \neq c_2$$

dcs

	a	b	c	d	e
a	3	2	2	1	1
c	2	2	2	1	1
d	1	1	1	1	1
e	1	1	1	1	1



```

for(int i = dp.length-2; i >= 0; i--) {
    for(int j = dp[0].length-2; j >= 0; j--) {
        if(s1.charAt(i) == s2.charAt(j)) {
            dp[i][j] = dp[i+1][j+1] + 1;
        }
        else {
            dp[i][j] = Math.max(dp[i+1][j], dp[i][j+1]);
        }
    }
}

```

S1: a b f c

S2: a k b c m

	a	k	b	c	m	-
a	3	2	2	1	0	0
b	2	2	2	1	0	0
f	1	1	1	1	0	0
c	1	1	1	1	0	0
-	0	0	0	0	0	0