Count Palindromic Substrings

a d b c c b a

et

st

	0 a	1 d	2 b	3 c	4 c	5 b	6 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

$i \ldots\ldots j$ → $ch(i) == ch(j)$

&& pal$(i+1, j-1)$

count : total no. T

longest : last true (diagonal)

Longest Palindromic Subsequences

str : a b c k b

brute force : 2^n

all palindromic subseq.

a　　b b　　$\boxed{b c b}$

b

c

b

lps

longest pal subseq.

ans : 3

$lps(str) = lps(c_1\, m\, c_2) = S(c_1\, m\, c_2)$

$c_1 \rightarrow$ left ext

$c_2 \rightarrow$ right ext

$= \_\ S(m)\ \_$ ①

$\_\ S(m)\ c_2$ ②

$c_1\ S(m)\ \_$ ③

$c_1\ S(m)\ c_2$ ④

$\underline{a}\ \underline{b\ x}\ \underline{d}$

$c_1\quad m\quad c_2$

$\boxed{\_\ S(m)\ \_}$	$\boxed{\_\ S(m)\ c_2}$
_ _ _ _	_ _ _ d
_ _ x _	_ _ x d
_ b _ _	_ b _ d
_ b x _	_ b x d

$\boxed{c_1\ S(m)\ \_}$	$\boxed{c_1\ S(m)\ c_2}$
a _ _ _	a _ _ d
a _ x _	a _ x d
a b _ _	a b _ d
a b x _	a b x d

$c_1 == c_2$	$c_1 \ != c_2$
④	①, ②, ③
$lps(s) = lps(m) + 2$	$lps(s) = \max(①, ②, ③)$
	$= \max(lps(c_1 m),\ lps(m c_2))$

$lps(c_1 m)$ → ①, ③

$lps(m c_2)$ → ①, ②

$- \; s(m) \; -$ ①

$- \; s(m) \; c_2$ ②

$c_1 \; s(m) \; -$ ③

$c_1 \; s(m) \; c_2$ ④

$c_1 == c_2$

a bkgekaj a

3 → a kgka

a kgk a

$lps(\underbrace{c_1\, m\, c_2}_{str})$ $= \begin{cases} lps(m) + 2 & c_1 == c_2 \\ \max\begin{pmatrix} lps(c_1 m), \\ lps(m c_2) \end{pmatrix} & c_1 \;!= c_2 \end{cases}$

$lps(m) \to i+1,\ j-1$

$lps(c_1 m) \to i,\ j-1$

$lps(m c_2) \to i+1,\ j$

string: abkccbc

$$\begin{tikzcd}
c_1 m & c_1 m c_2 \arrow[l] \arrow[dl] \arrow[d] \\
m & m c_2
\end{tikzcd}$$

	a_0	b_1	k_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
k_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	1
c_6	X	X	X	X	X	X	1

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

Count Palindromic Subsequences

a b c k b

a

b

c

k

b

b b

b c b

b k b

count = 8

Count Palindromic Subsequences

$cps(str) = cps(c_1\ m\ c_2) = s(c_1\ m\ c_2)$

$= \ -\ s(m)\ -$ ①

$-\ s(m)\ c_2$ ②

$c_1\ s(m)\ -$ ③

$c_1\ s(m)\ c_2$ ④

$\boxed{④ = ① + \underbrace{1}_{c_1c_2}}$

$c_1 == c_2$	$c_1 \,!= c_2$
① + ② + ③ + ④	① + ② + ③
$cps(str) = ① + ② + ③ + ④$ $= \underbrace{① + ②}_{cps(m c_2)} + \underbrace{③ + ①}_{cps(c_1 m)} + 1$ $cps(m c_2) + cps(c_1 m) + 1$	$cps(str) = \underbrace{① + ②}_{cps(m c_2)} + \underbrace{③ + ①}_{cps(c_1 m)} - \underbrace{①}_{cps(m)}$ $= cps(m c_2) + cps(c_1 m) - cps(m)$

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

$$\begin{tikzcd} c_1 m & c_1 m c_2 \arrow[l] \arrow[dl] \arrow[d] \\ m & m c_2 \end{tikzcd}$$

	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

abd

lcs

$lcs(s1, s2) = lcs(c1r1, c2r2)$

$= s(c1r1) \times s(c2r2)$

$= \begin{bmatrix} -s(r1) \\ c1\,s(r1) \end{bmatrix} \times \begin{bmatrix} -s(r2) \\ c2\,s(r2) \end{bmatrix}$

c1 → starting char of s1

r1 → remaining str of s1

c2 → starting char of s2

r2 → remaining str of s2

abd

$-s(r1)$	$\boxed{\text{\_}, d, b, bd}$
$c1\,s(r1)$	$\boxed{a, ad, ab, abd}$

bcd

$-s(r2)$	$\boxed{\text{\_}, d, e, ed}$
$c2\,s(r2)$	$\boxed{b, bd, be, bed}$

$= -s(r1) \times -s(r2)$ (1)

$-s(r1) \times c2\,s(r2)$ (2)

$c1\,s(r1) \times -s(r2)$ (3)

$c1\,s(r1) \times c2\,s(r2)$ (4)

c1 == c2	c1 != c2
(4)	(1), (2), (3)
$lcs(s1,s2) = lcs(r1,r2) + 1$	$lcs(s1,s2) = \max\left(\underbrace{lcs(c1r1, r2)}_{(1),(3)}, \underbrace{lcs(r1, c2r2)}_{(1),(2)}\right)$

$$lcs(s1, s2) = lcs(c1r1, c2r2) = \begin{cases} lcs(r1, r2) + 1 & c1 == c2 \\ \max(lcs(c1r1, r2),\ lcs(r1, c2r2)) & c1 \,!= c2 \end{cases}$$

lcs

i,j

	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

$$(c1r1,\ c2r2) \to c1r1,\ r2$$

$$(c1r1,\ c2r2) \to r1,\ r2$$

$$(c1r1,\ c2r2) \to r1,\ c2r2$$

```
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 j 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
j	1	1	1	1	0	0
c	1	1	1	1	0	0
_	0	0	0	0	0	0