Problem Definition:

給定一 string 為 s，找出 s 中最長的 Palindromic substring 為何?

Example: ①. s = "babad"

則 最長的 Palindrome substring 可為:

"bab" 或 "aba".

②. s = "cbbd"

則 最長的 Palindrome substring 為:

"bb"

Solution:

① 遍歷所有 substring，check 每一 substring 是否為 Palindrome.

⇒ Time: $O(n^2) \times O(n) = O(n^3)$

② 是否有 overlapping subproblems 且可用 subproblems 得到原 problem 的解.

⇒ Dynamic Programing.

Dynamic Programming:

令 $d_{i,j}$ 為考慮 substring $s[i:j]$ 時的最長 Palindrome 長度。

則可知：$d_{i,i} = 1$，$\forall i = 1, \dots, n$

而 $d_{i,i+1} = \begin{cases} 2 & s[i] == s[i+1] \\ 0 & \text{otherwise} \end{cases}$

Example:

① $s =$ "babad"

	b	a	b	a	d
b	1	0	3		
a	✓	1	0	3	
b	X	X	1	0	0
a	X	X	X	1	0
d	X	X	X	X	1

← $d_{1,3}$

② $s =$ "abba"

	a	b	b	a
a	1	0	0	
b	X	1	2	0
b	X	X	1	0
a	X	X	X	1

Define recursive function:

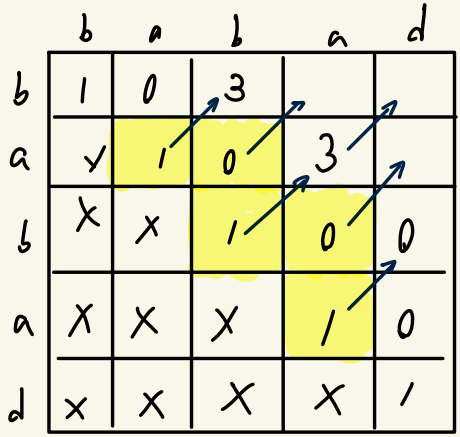
$$d_{i,j} = \begin{cases} d_{i+1,j-1} + 2 & \text{if } s[i] == s[j] \text{ and } d_{i+1,j-1} \neq 0 \\ 0 & \text{otherwise} \end{cases}$$

Optimization:

We can observe that when we calculate the $d_{i,j}$ we only need the $d_{i+1,j-1}$

e.g.

$s =$ "babad"



②. s = " c b b d "

	c	b	b	d
c	1	0		
b		1	2	
b			1	0
d				1

what we need to store :

②. s: " a b c c b a "

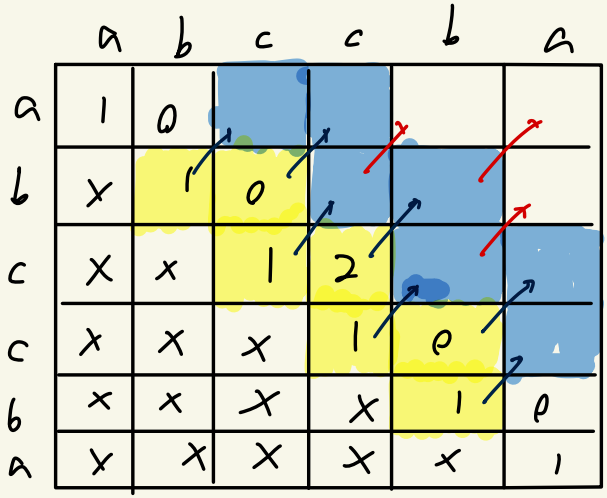
	a	b	c	c	b	a
a	1	0				
b	x	1	0			
c	x	x	1	2		
c	x	x	x	1	e	
b	x	x	x	x	1	e
a	x	x	x	x	x	1

What would be the table size?

$\Rightarrow$ size = (len − 2) × 2 − 1

How to calculate the table index?

r = "a b c c b a"



1	0	1	2	1	0	1

left	0	0	1	1	2	2	3
right	2	3	3	4	4	5	5

left	0	0	1
right	4	5	5

⇒ index = (left + right) - 2

Example: "abcdbbfcba"

	a	b	c	d	b	b	f	c	b	a
a	1	0								
b		1	0							
c			1	0						
d				1	0					
b					1	2				
b						1	0			
f							1	0		
c								1	0	
b									1	0
a										1

I.

Note: When we count the optimal solution, don't forget the edge case:

r: " a b c c b a "

	a	b	c	c	b	a
a	1	0				
b	x	1	0			
c	x	x	1	2		
c	x	x	x	1	0	
b	x	x	x	x	1	0
a	x	x	x	x	x	1

: edge case

II.

Hard to implement the traverse process.

Solution 3: ① 遍歷每个字元作為 center，往外 expand，找最長的作為 substring.

∃ 要注意 Palindrome substring 有兩种 case

① 偶數長度，eg. "aabb"

② 奇數長度，eg. "cbabc"

expand 時，要考慮此二种情形

可在字元間隔插入 - delimiter 来作為捨棄偶數長度 Palindrome substring 輔助.

e.g. "aabb" ⇒ "*a*a*b*b*"

插入 delimiter 後，必可保證原字串長度為奇數.

Why? ∵ 設 s 中有 n 个 char，則共有 n+1 个間隔

∴ 插入 delimiter 後為 $2n+1$ 個字元.

"ababa" ≠ "*a*b*a*b*a"