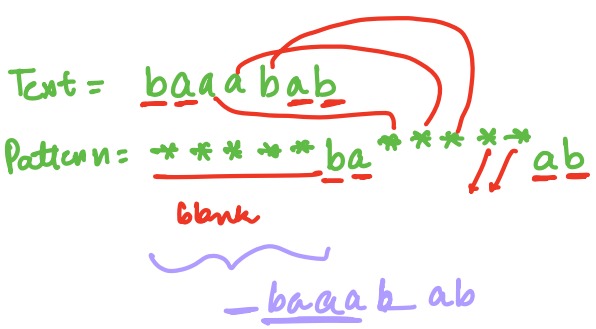
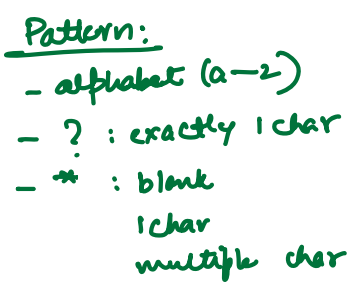
Wildcard Pattern Matching

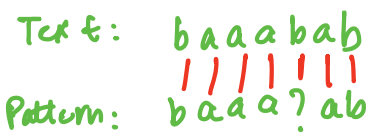
Given a text and a wildcard pattern, implement wildcard pattern matching algorithm that finds if wildcard pattern is matched with text. The matching should cover the entire text (not partial text).
The wildcard pattern can include the characters '?' and '*'
'?' – matches any single character
'*' – Matches any sequence of characters (including the empty sequence)

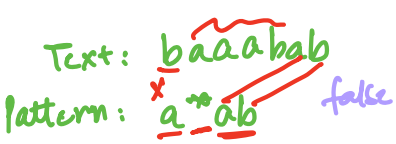
For example,
Text = "baaabab",
Pattern = "*****ba*****ab", output : true
Pattern = "baaa?ab", output : true
Pattern = "ba*a?", output : true
Pattern = "a*ab", output : false

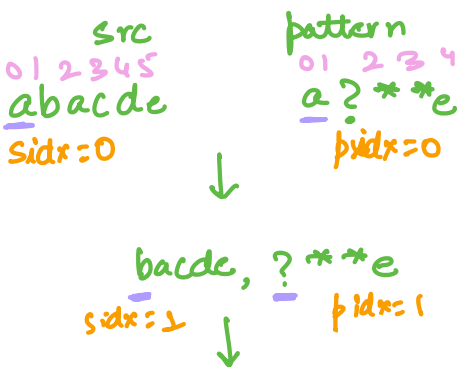
Each occurrence of '?' character in wildcard pattern can be replaced with any other character and each occurrence of '*' with a sequence of characters such that the wildcard pattern becomes identical to the input string after replacement.

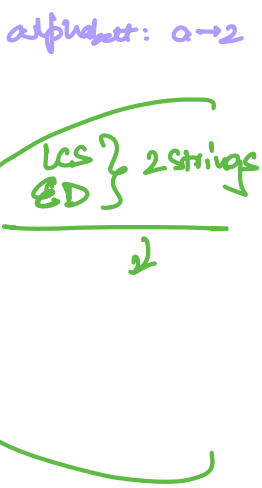


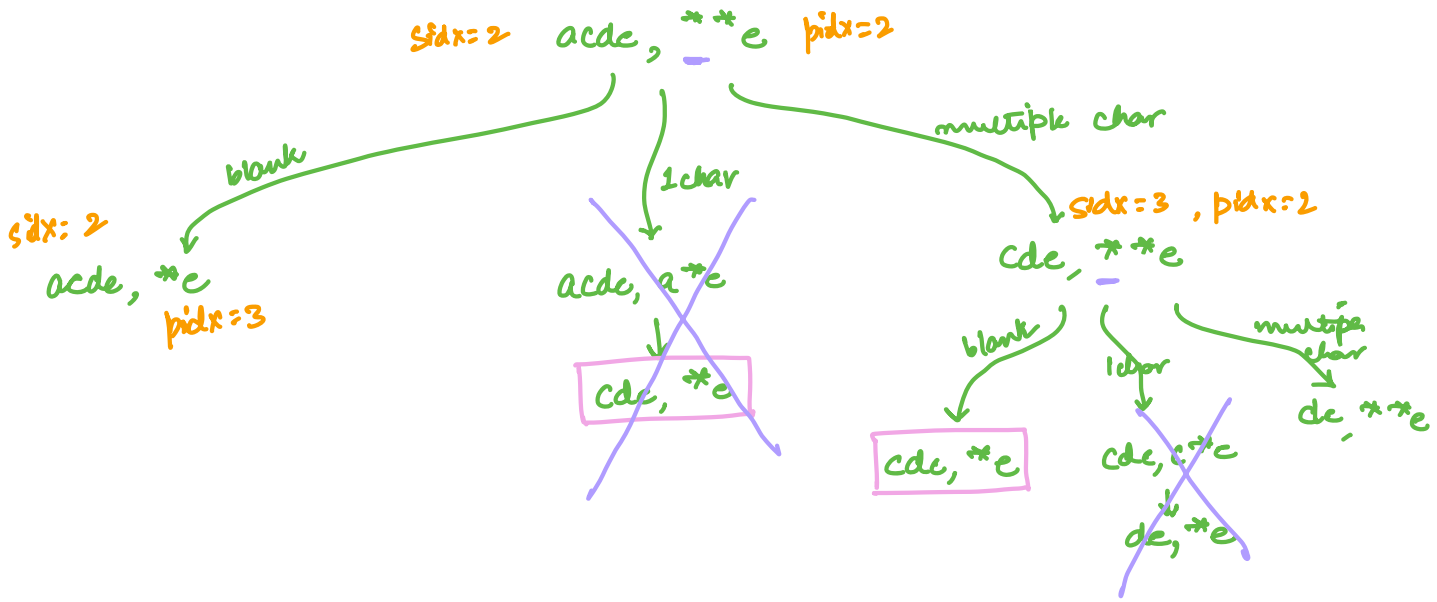












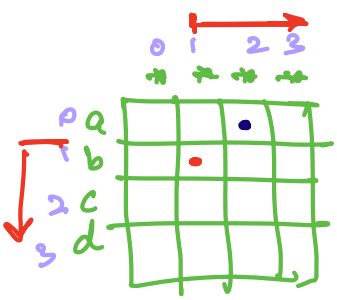
Text = baaab

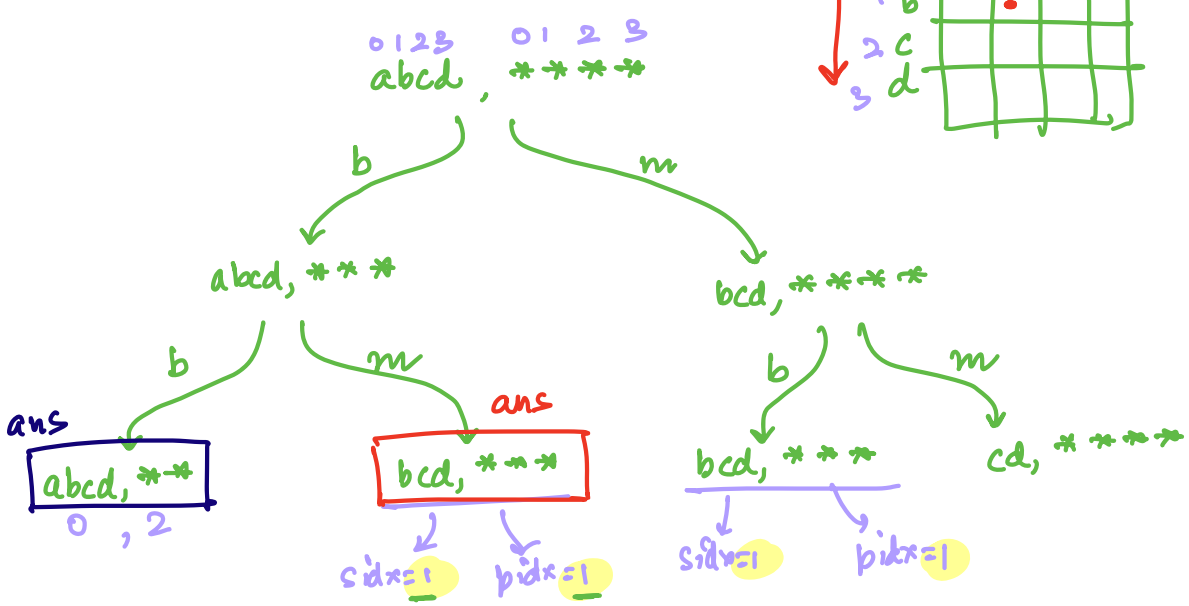
Pattern : * ab ?

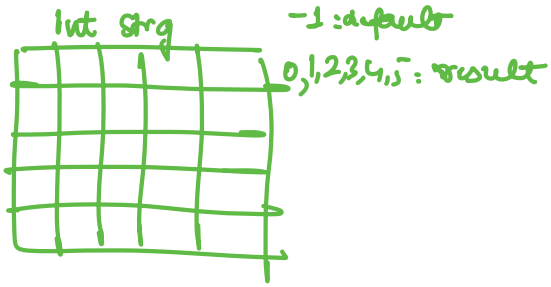
Base Case:

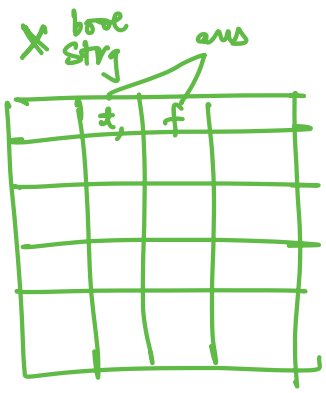
	src	pat	
①	—	—	: true
②	abc	—	: false
③	—	?ab	: false
	—	****	: true

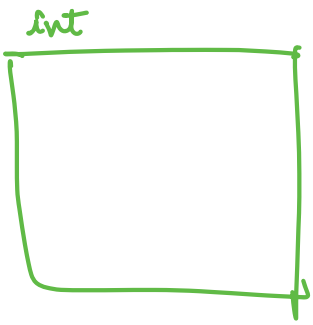
pidx → length left











Bottom Up:

- 2D
- meaning
- TD BC ↘
  BU fill

last row fill:

4,2 cell

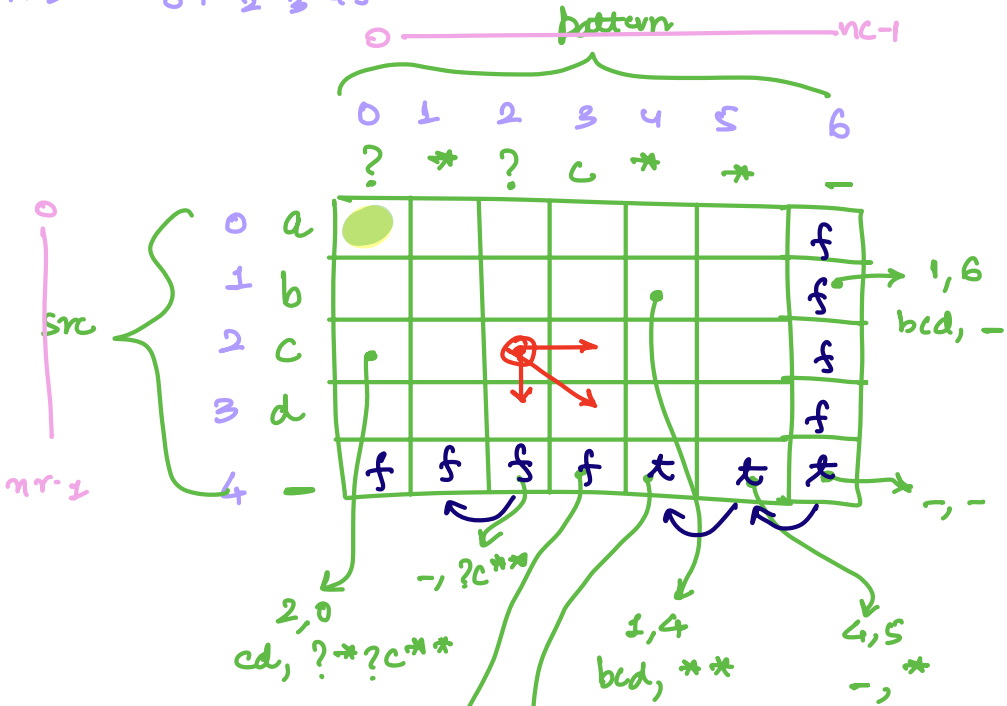
—, ?c** : false

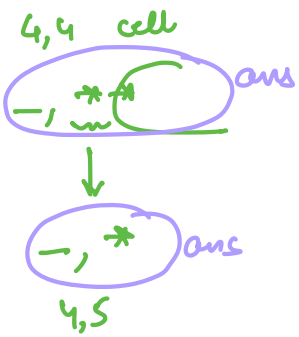
4,3 cell

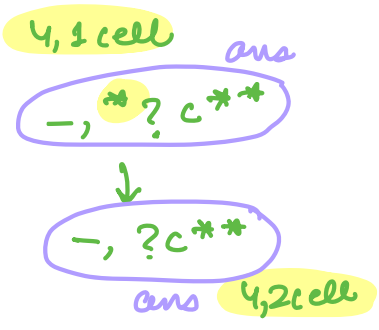
—, c** : false

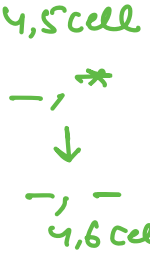
src: abcd (0 1 2 3)

pat: ?*?c** (0 1 2 3 4 5)





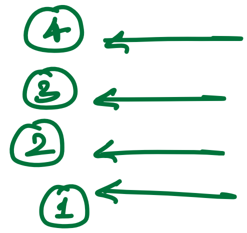




last row:

```
if (chp == '*')
    strg[row][col] = strg[row][col+1];
else
```

strg[rows][cols]: false;



	?	*	?	c	*	*	_
a	t	t	f	f	t	t	f
b	f	t	t	f	t	t	f
c	f	f	f	t	t	t	f
d	f	f	f	f	t	t	f
_	f	f	f	f	t	t	t

Palindrome Partitioning

Rec:

nitin → n | i | t | i | n → 4

→ n | iti | n → 2

→ nitin → 0

Answers Print

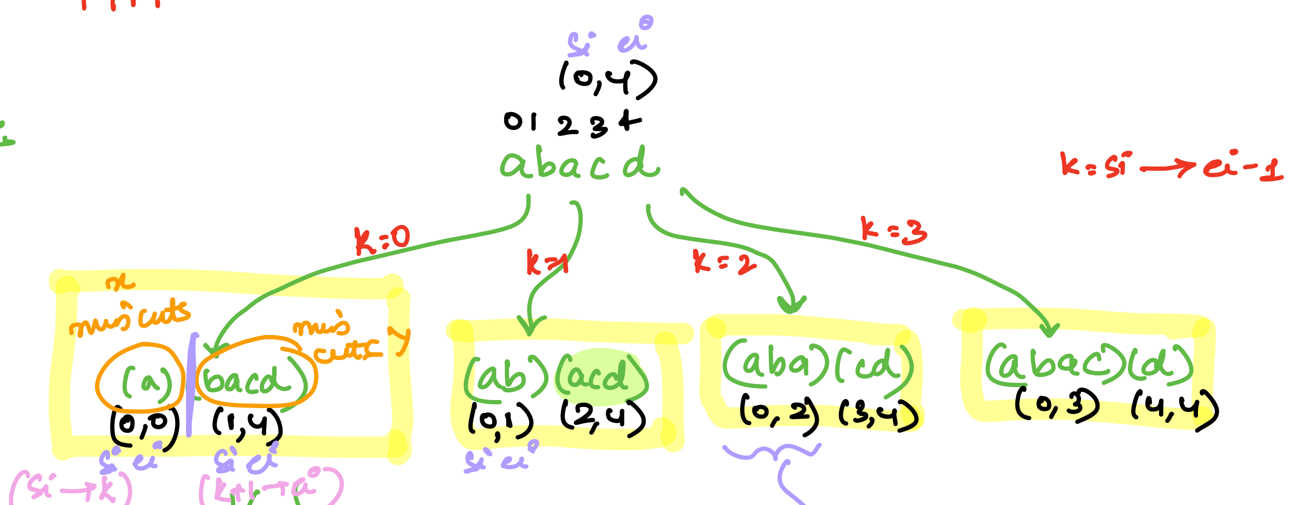
Ques: aba|cdc|e

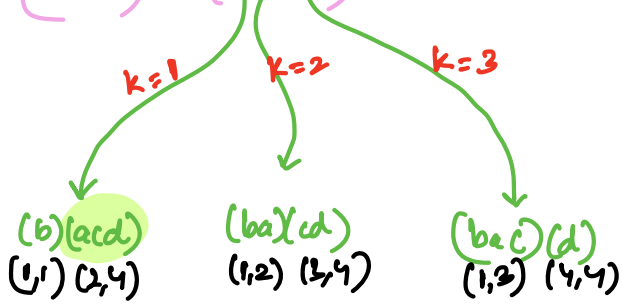
↳ ans: 2

min cuts such that each part is a palindrome

WC: a|b|c|d|c length() - 1 cuts.

Recursively:



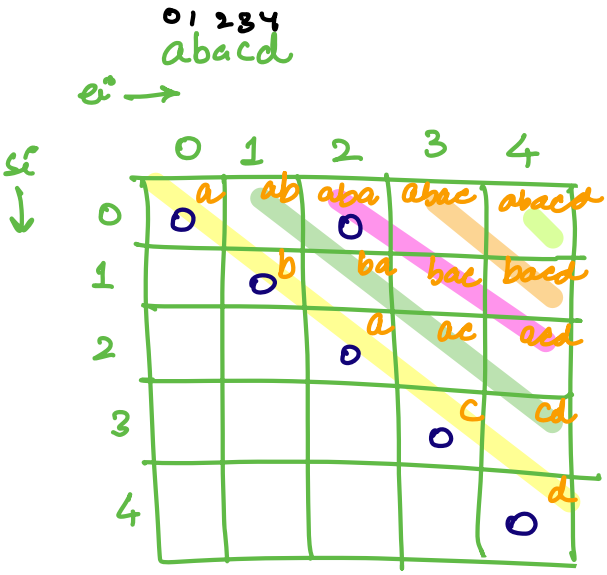


Palindrome
bc
return 0

n a t b n

Bottom up:

- 2D strg
- Cell meaning (min cuts)
- TD BC ⇓ BU fill } Copy
- filling dirn



TC: $\frac{n^2}{2} \times n$

$= O(n^3)$

5 length
↓
4 length
↓
3 length
↓
2 length

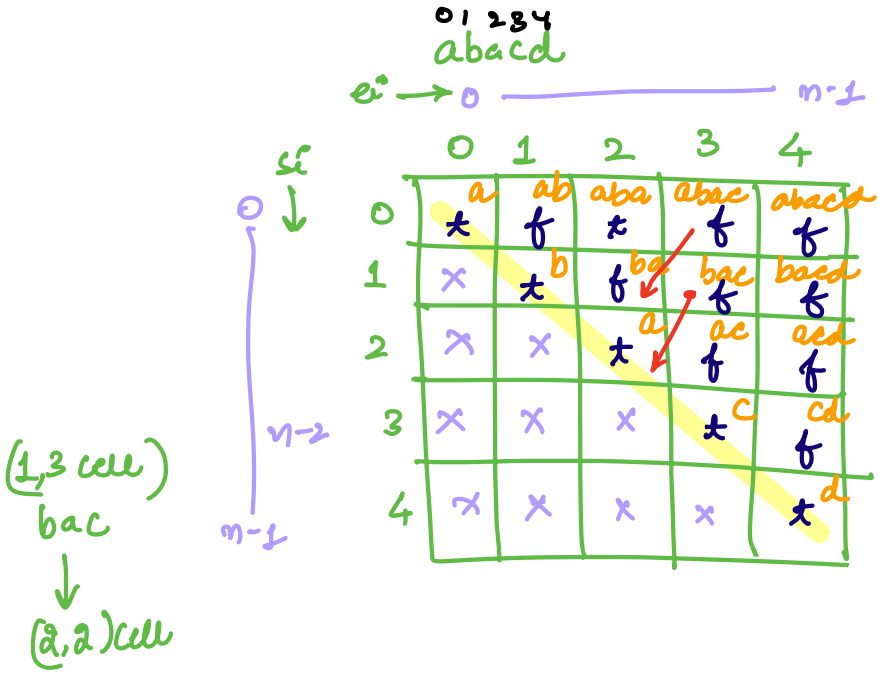
ei

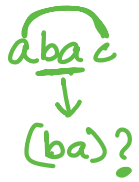
00	01	02	03	04
11	12	13	14	
22	23	24		
33	34			
44				
slide:0	slide:1	slide:2	slide:3	slide:4

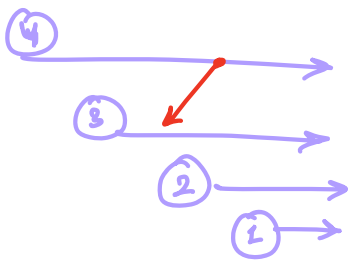
side	si	n-slide	n-slide-1
0	4	5	4
1	3	4	3
2	2	3	2
3	1	2	1
4	0	1	0

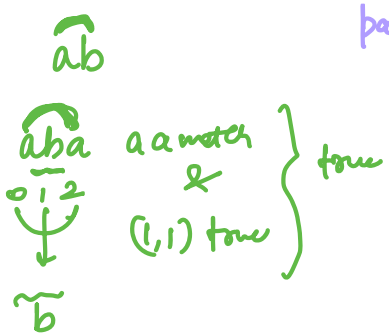
s-slide

Palindrome Info store:









HW: Palindrome Partitioning without & loop