



2015 JOI春合宿 Day3

AAQQZ(AAQQZ) 解説

2015/03/22
山下 洋史
@utakiyoshi



概要

- 文字列が与えられる
- ある区間をソートする
- 回文を探す
- うまい区間を見つけて最長の回文を作る
- 文字の種類数 = 長さ $\leq 3\ 000$

例



ZQQQAZAQSTAN

例



ZQQQAZAQSTAN

区間を選ぶ ZQQ**QAZAQ**STAN

例



ZQQQAZAQSTAN

区間を選ぶ ZQQ**QAZAQ**STAN

ソートする ZQQ**AAQQZ**STAN

例



	ZQQQAZAQSTAN
区間を選ぶ	ZQQQ AZA QSTAN
ソートする	ZQQ AAQQZ STAN
回文発見	ZQQAAQQZ STAN



小課題1

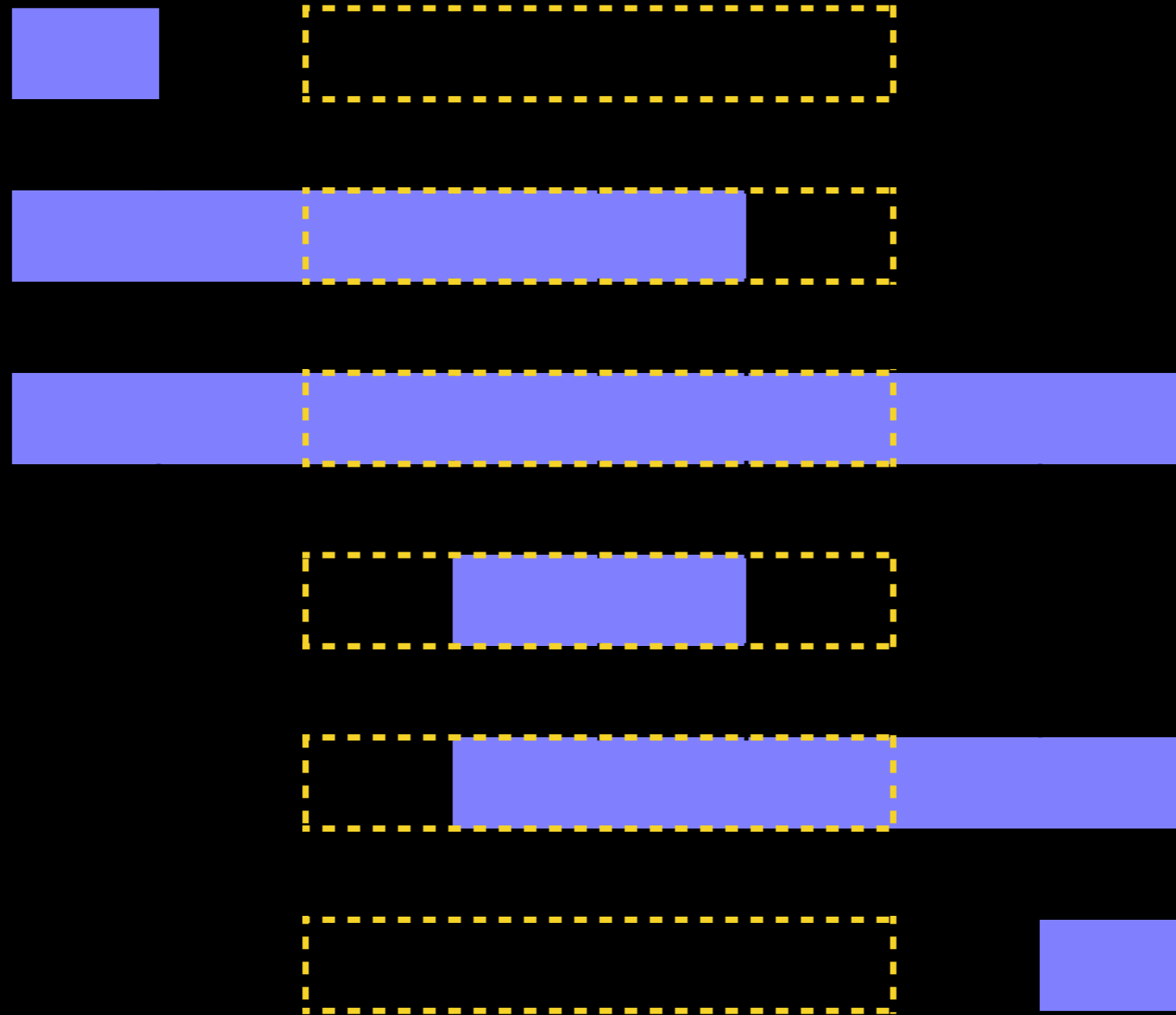
- $O(N^4)$ ($N \leq 50$)
 - 区間を $O(N^2)$ 通り試す
 - ソートして, $O(N^2)$ かけて最長回文を探す
- $O(N)$ の最長回文アルゴリズムなど使うと $O(N^3)$ になるけど.....
 - $N = 3\ 000$ は解けそうにない
 - 「Manacher のアルゴリズム」各自調べること
 - (例えば <http://snuke.hatenablog.com/entry/2014/12/02/235837>)



ここから本題 (満点解法)

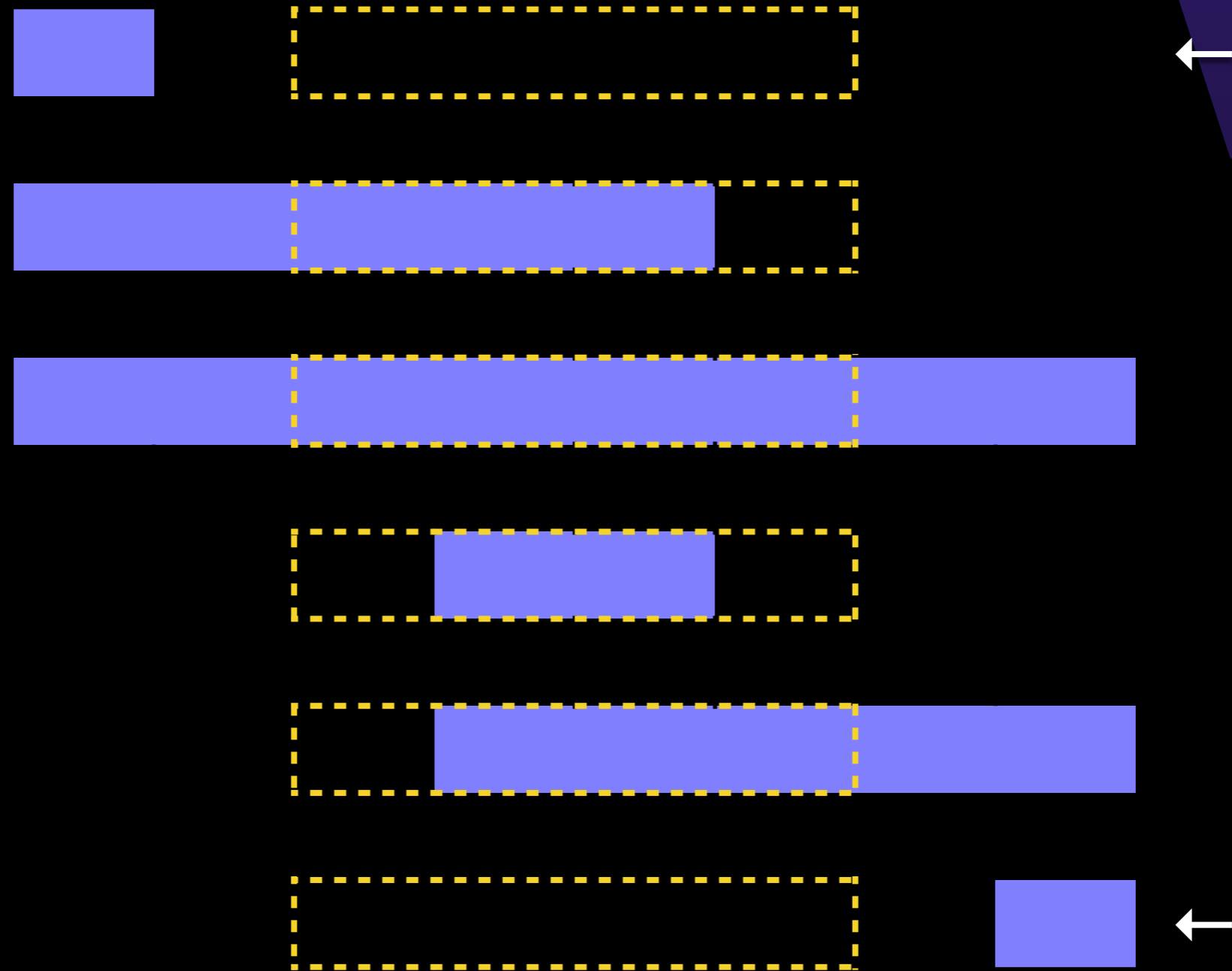


ソート列の 左右の端の位置で分類



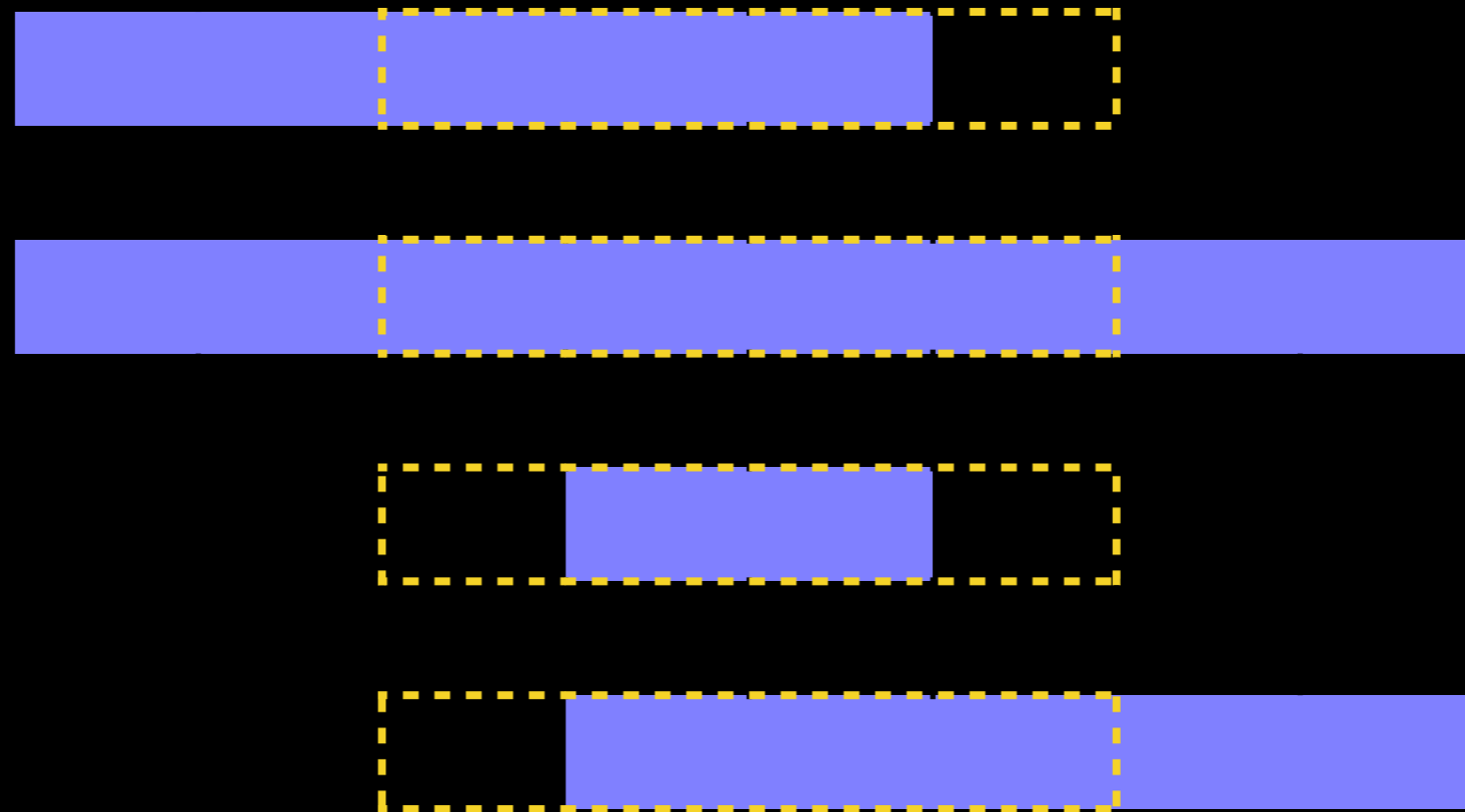


ソート列が回文の 外側に来るものは不要



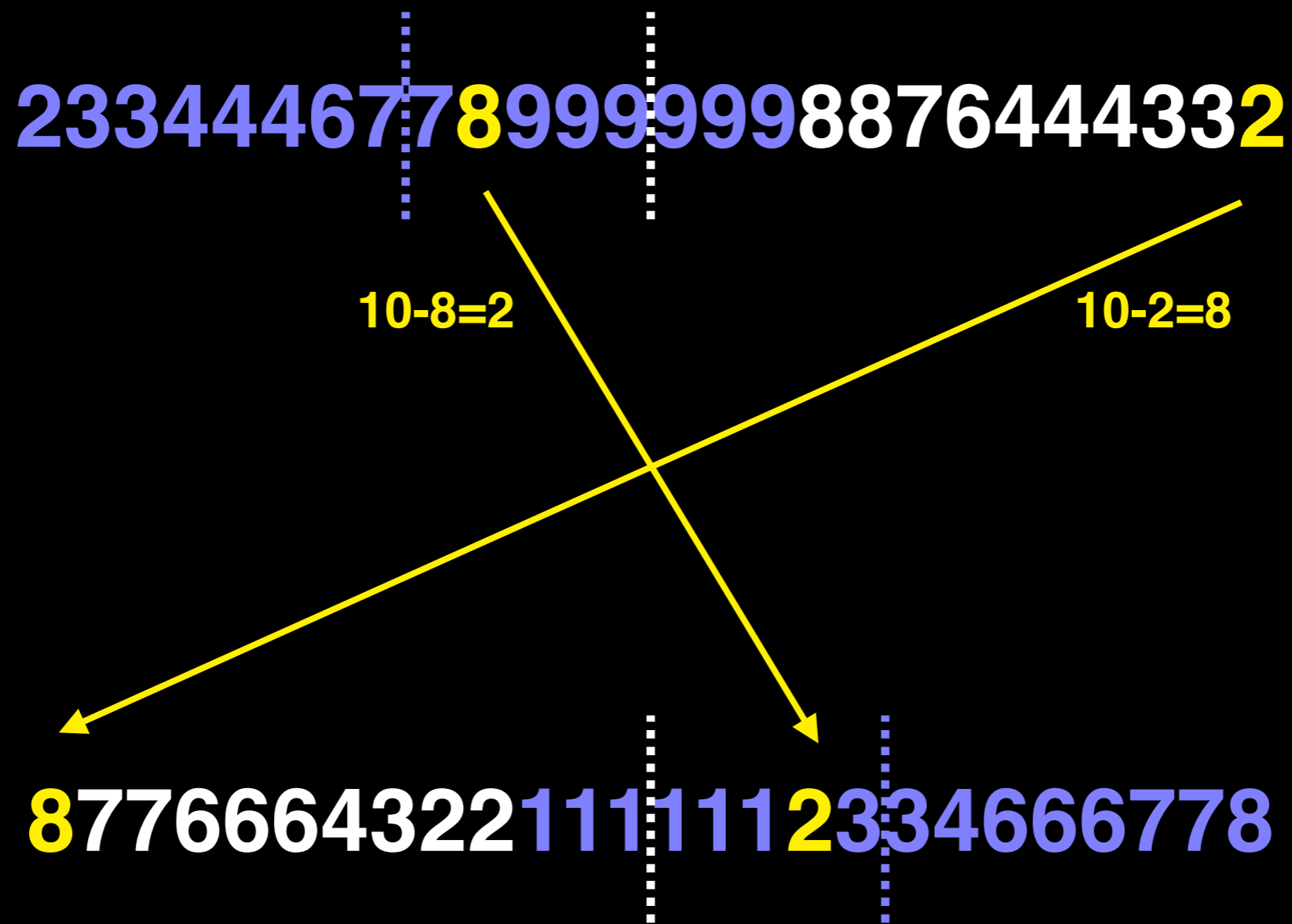


ソート列が回文の 外側に来るものは不要





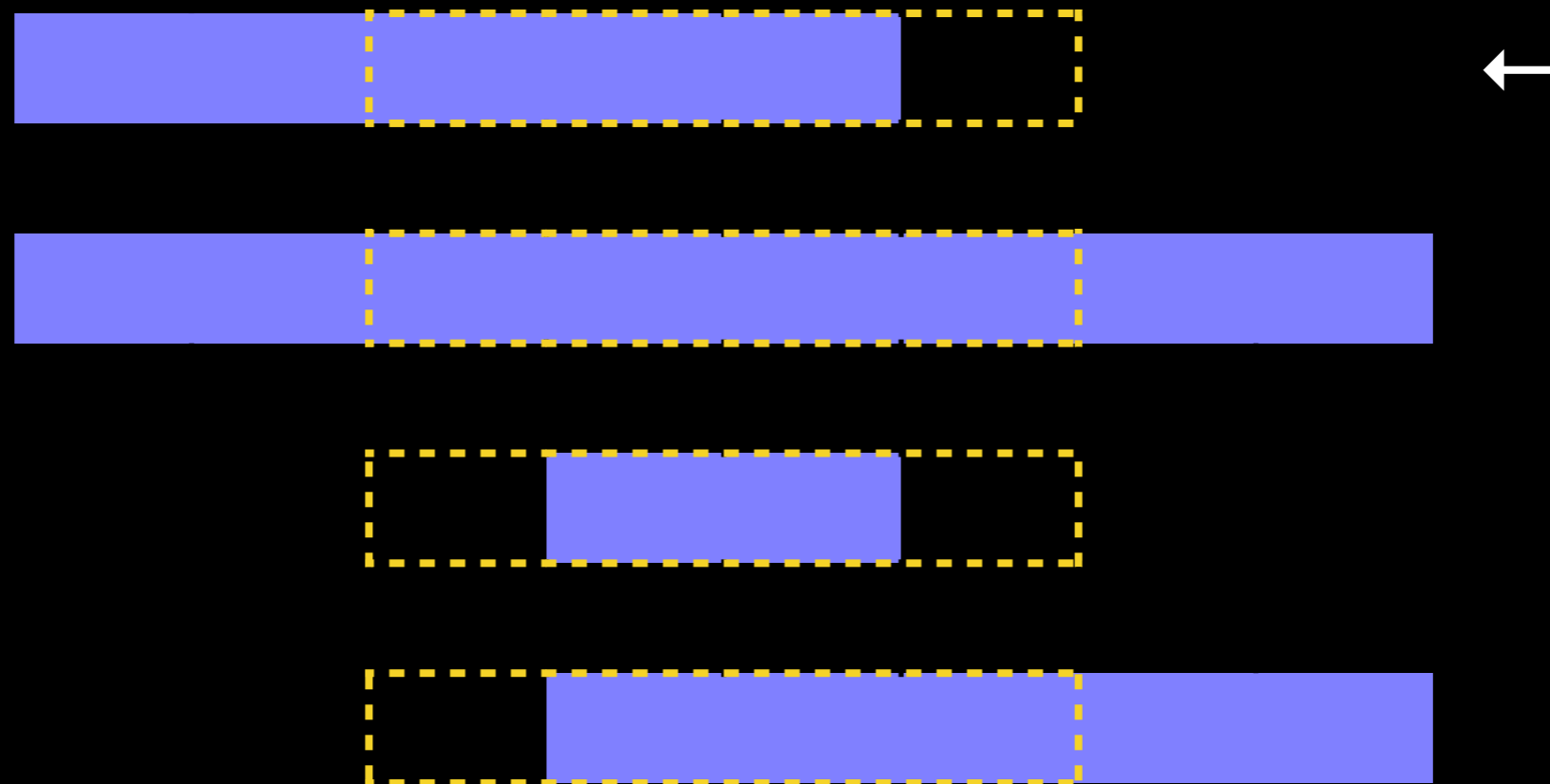
「反転」したものと
合わせて2回解く



ソート列の中心が回文の中心より右の場合だけ考えれば良い

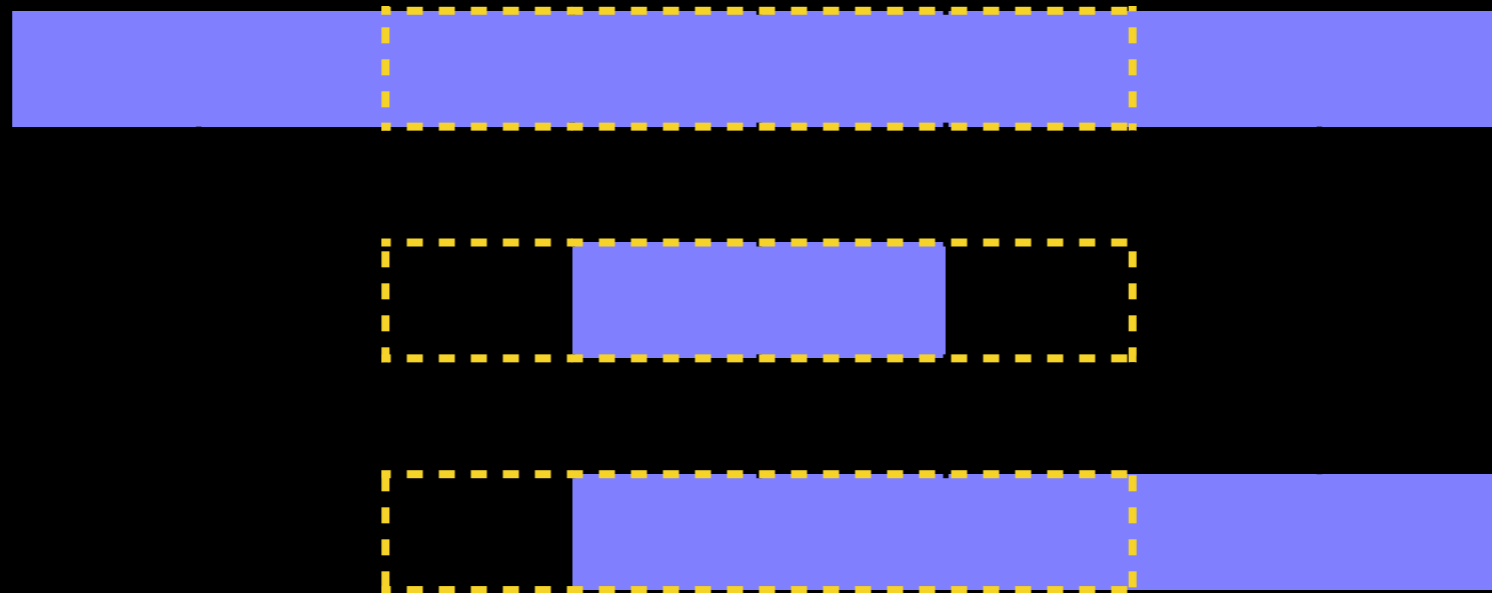


ソート列の中心が
回文の中心より左にあるもの



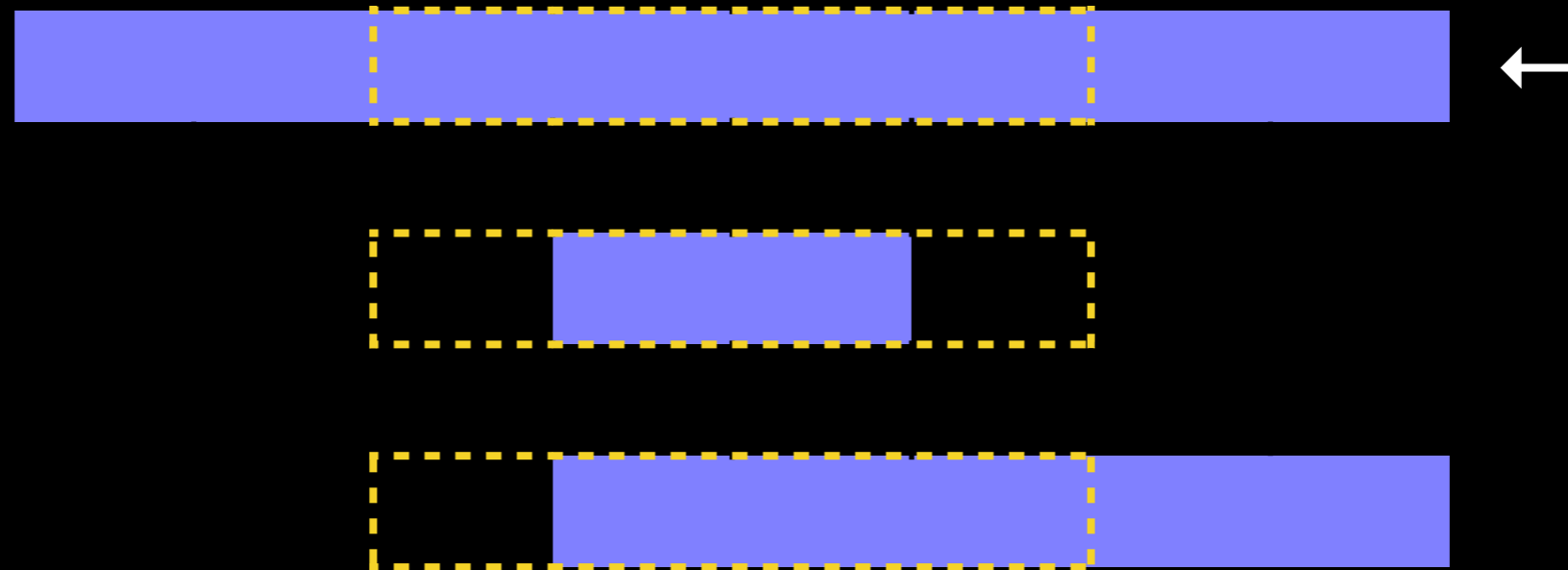


ソート列の中心が
回文の中心より左にあるもの



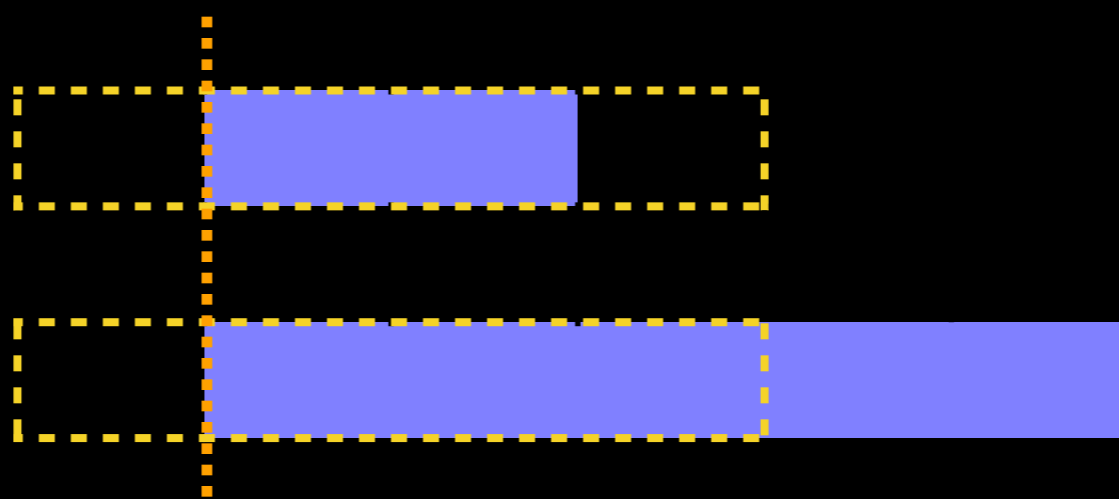


ソート列が回文を覆うものは後回し





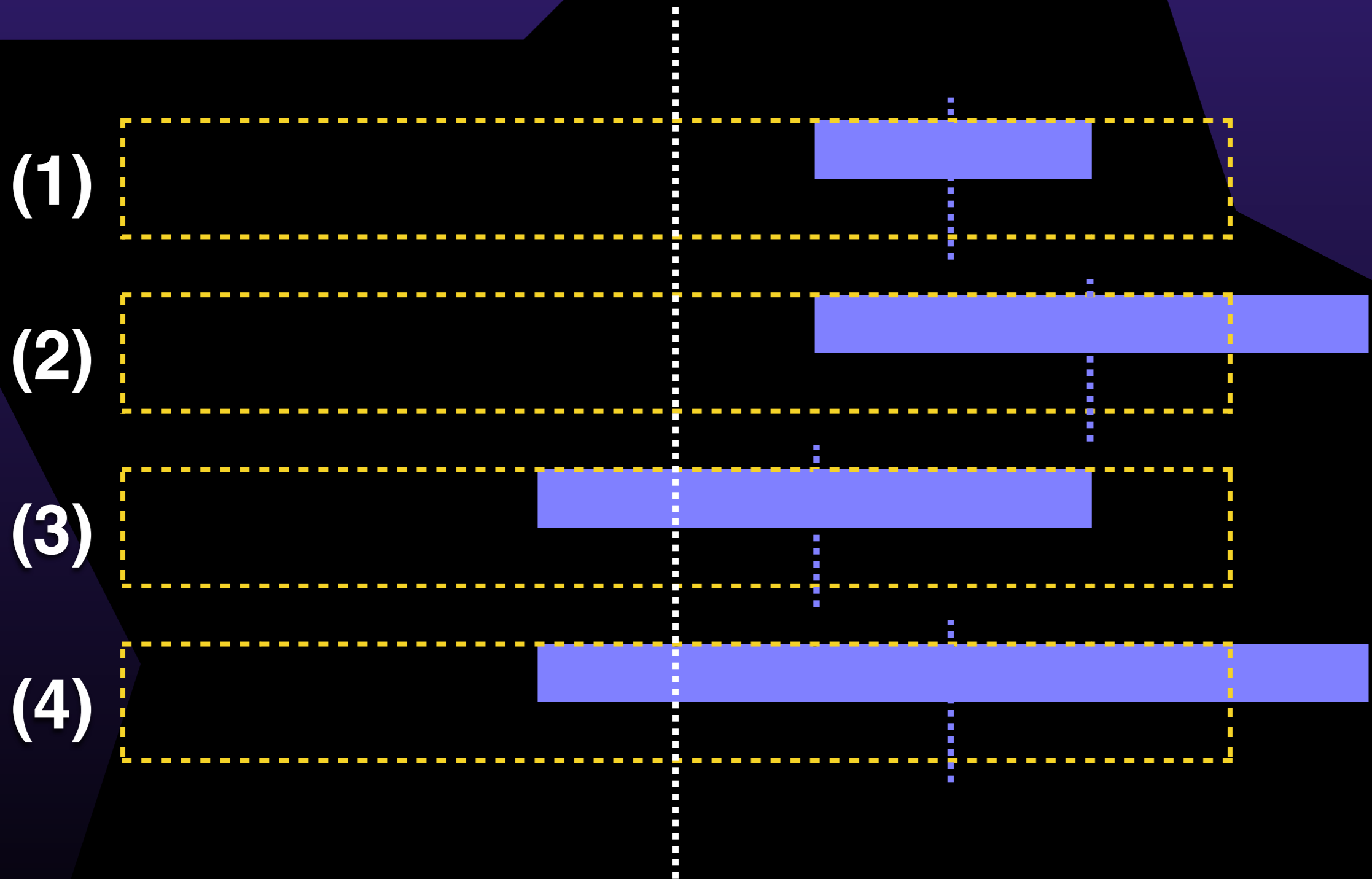
なので、
いま考えるべきなのは



ソート列の左端が回文内に入っているパターン

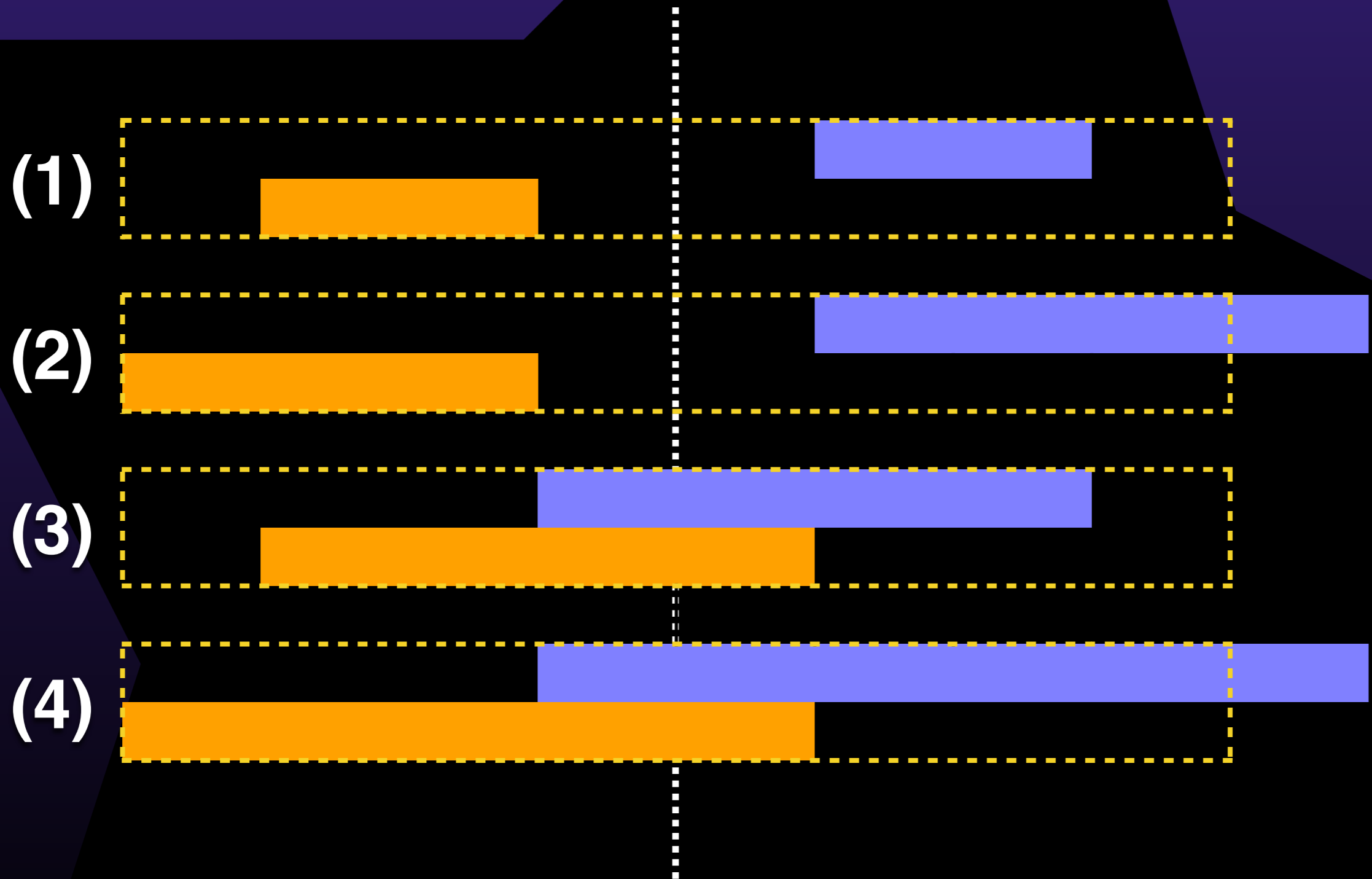


ソート列の 左右の端の位置で分類





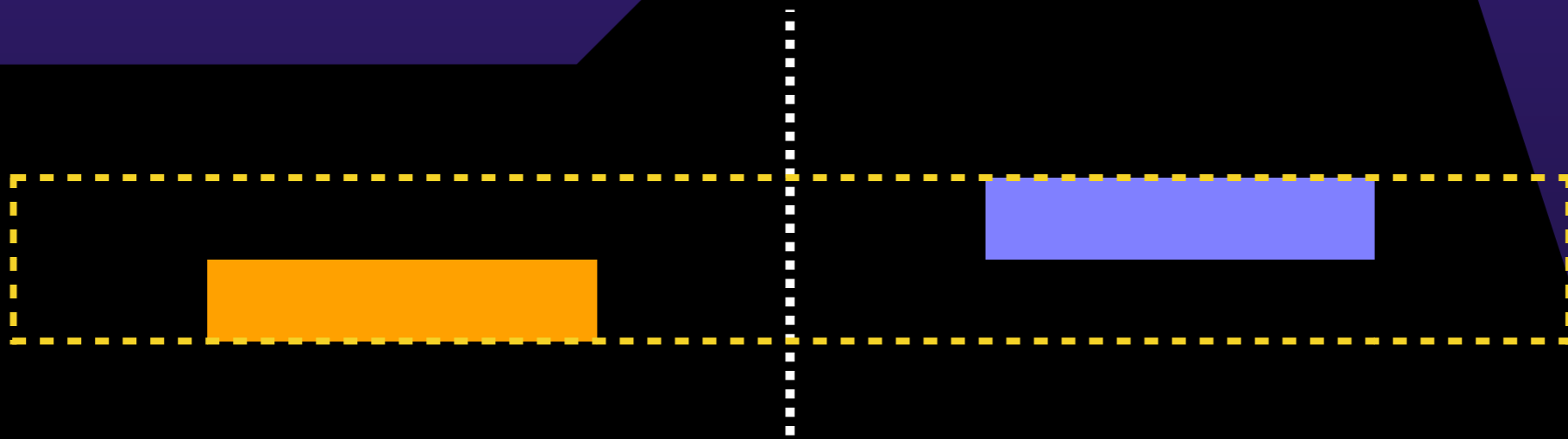
中心に関して対称に





パターン(1)

(1)

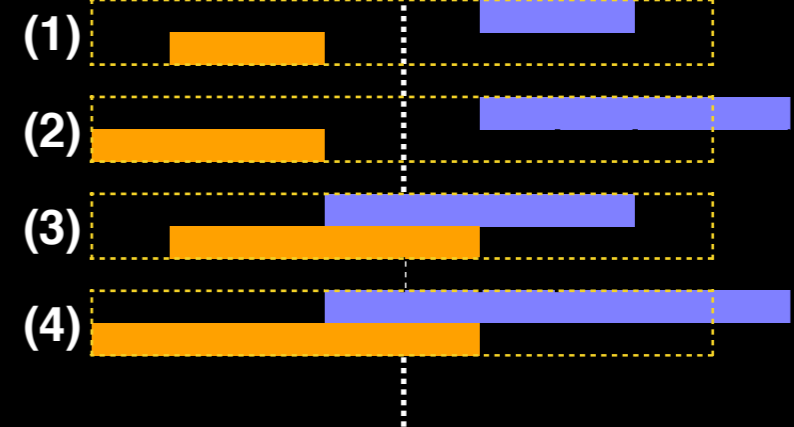


114 664332 514415 233466 411

降順

回文

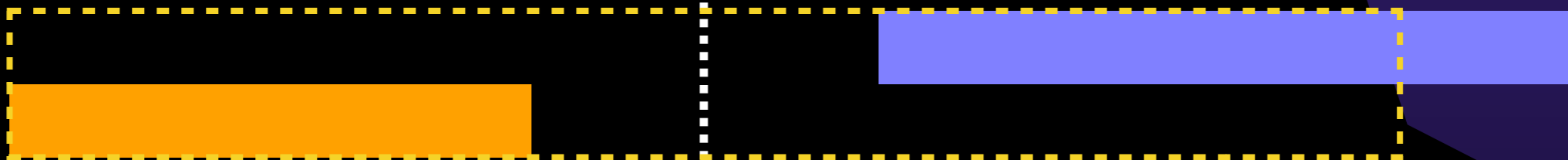
昇順





パターン(2)

(2)

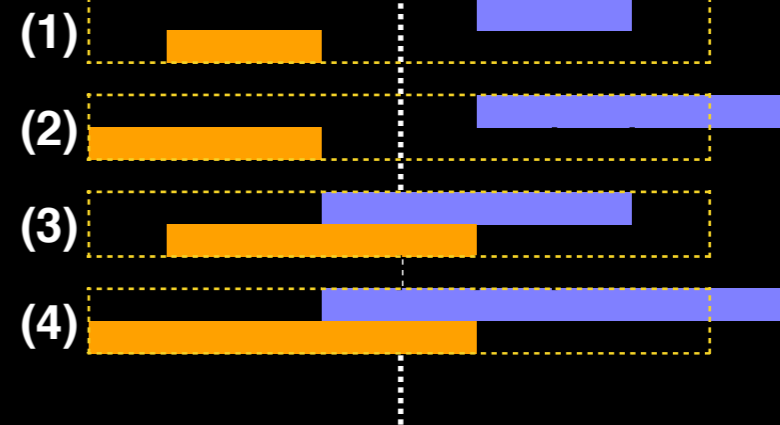


8776664332 514415 2334666778 899

降順

回文

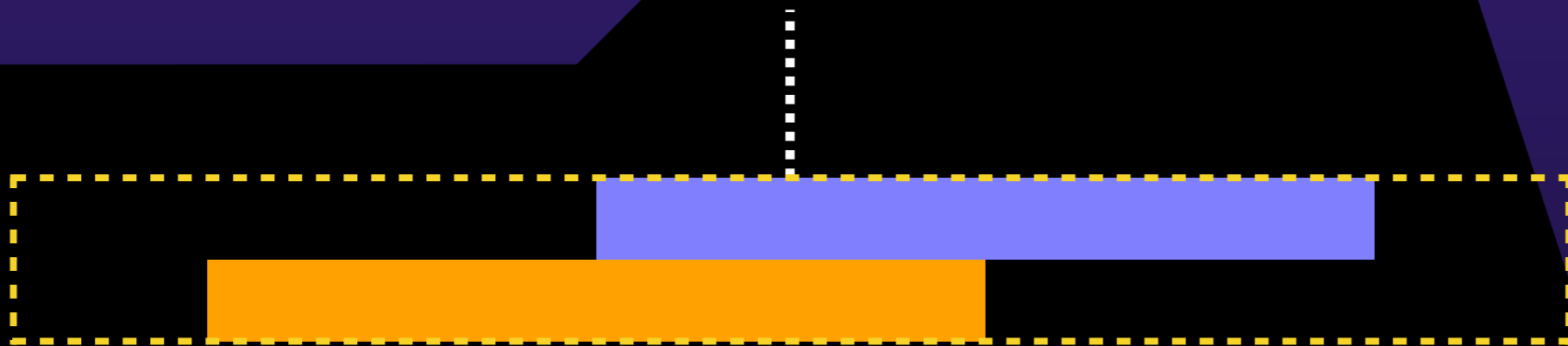
昇順





パターン(3)

(3)



114 664332 111111 233466 411

降順

一定

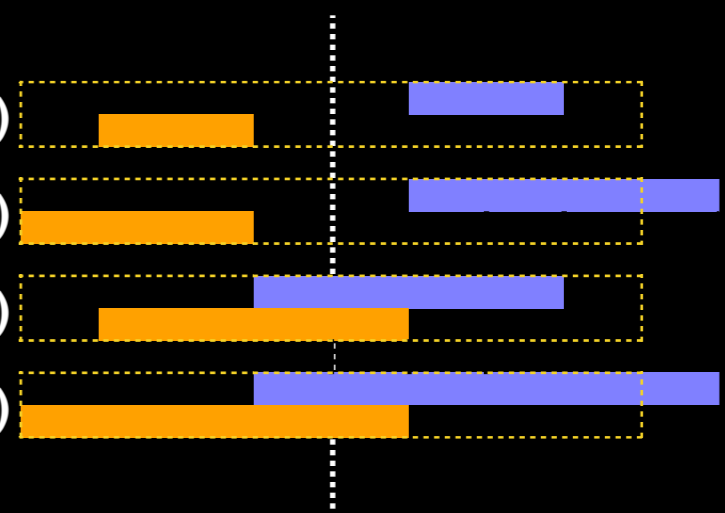
昇順

(1)

(2)

(3)

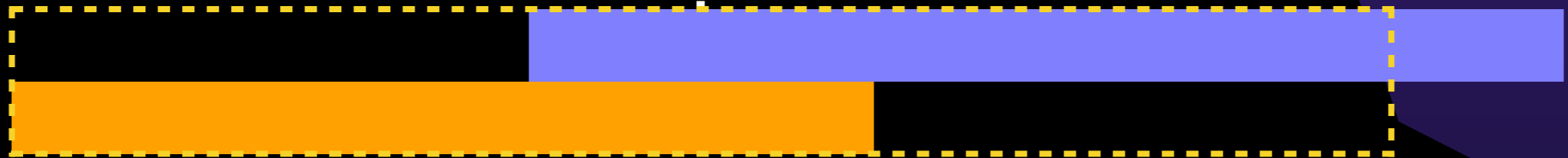
(4)





パターン(4)

(4)



8776664332

11111

2334666778 899

降順

一定

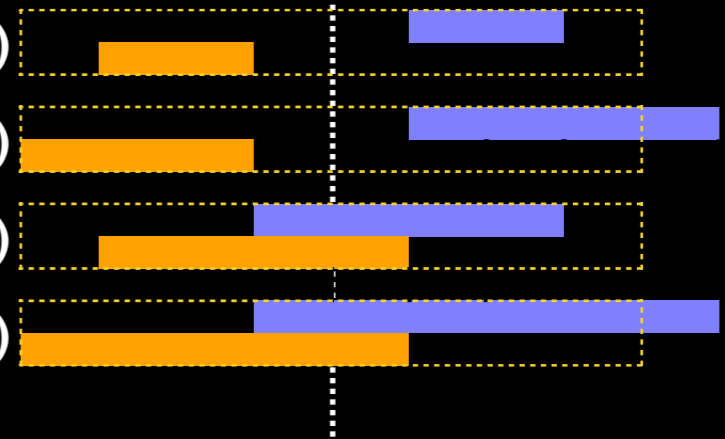
昇順

(1)

(2)

(3)

(4)





パターンのまとめ

(1) 114 664332 514415 233466 411

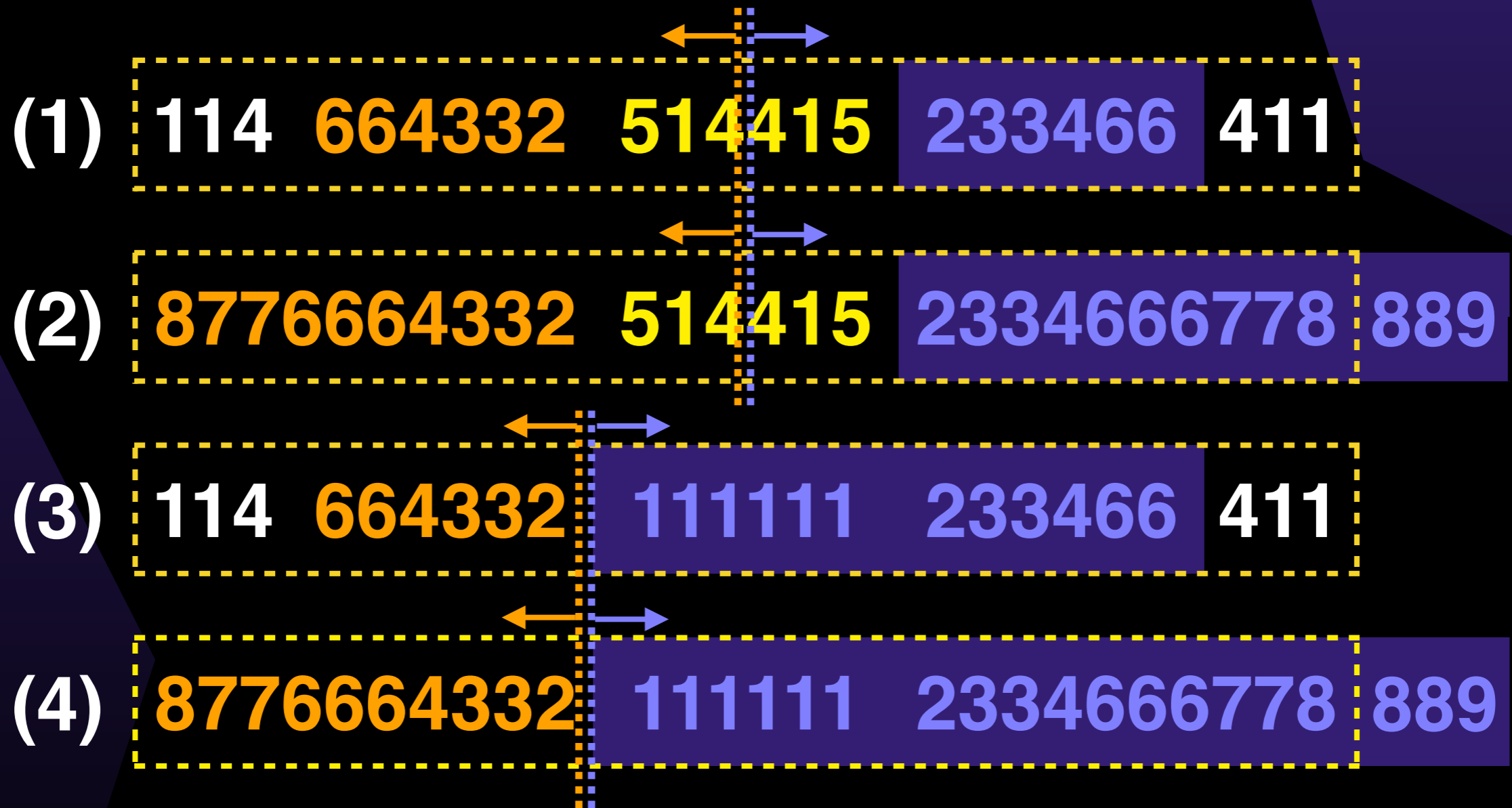
(2) 8776664332 514415 2334666778 889

(3) 114 664332 111111 233466 411

(4) 8776664332 111111 2334666778 889



開始位置をO(N)通り試す

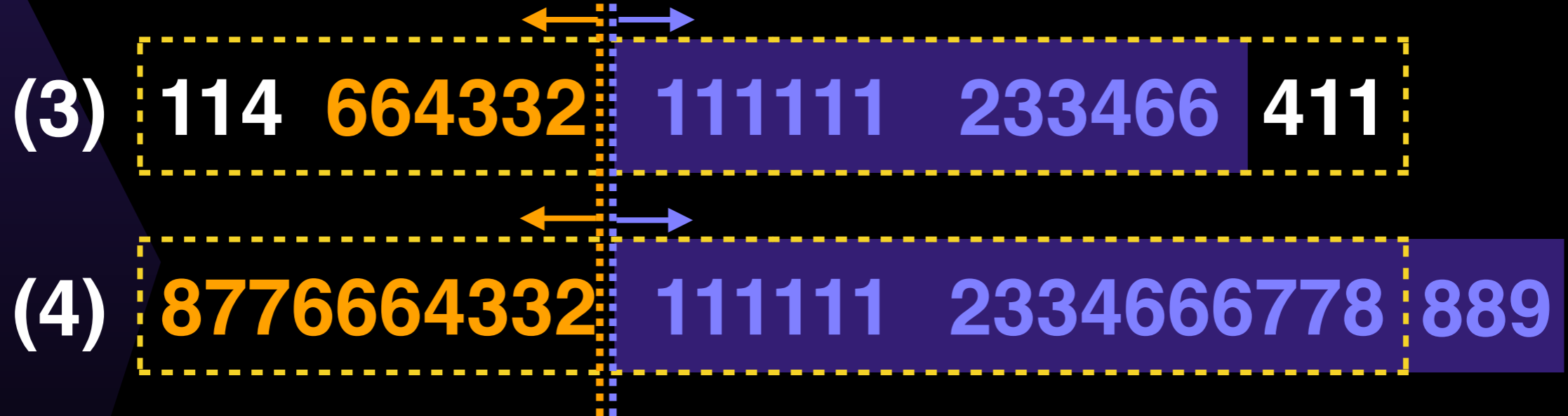




(1)(2)はまず
できるだけ長い回文をとる



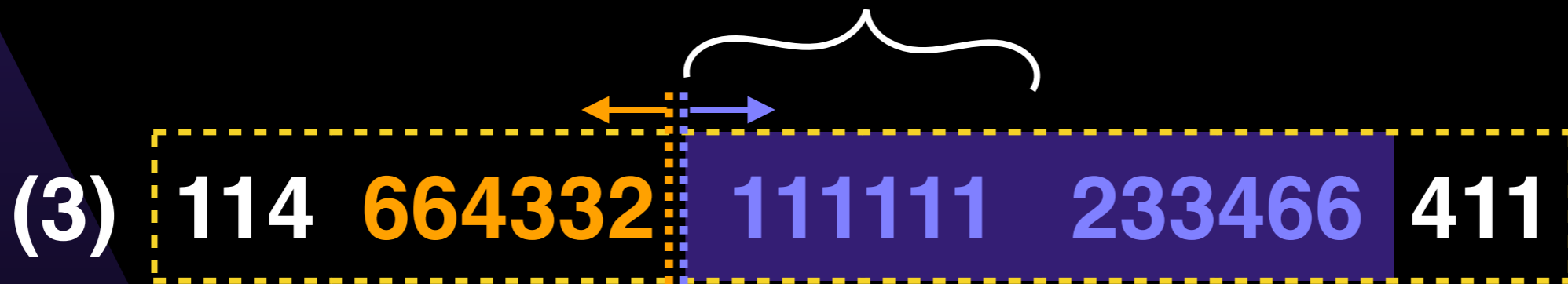
(3)(4)は？





(3)(4)は？

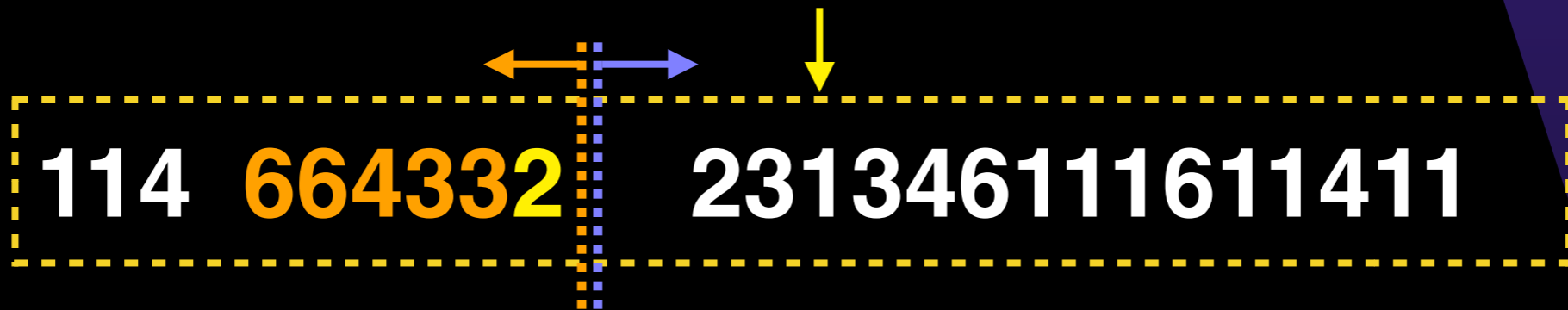
中心はどんな数？



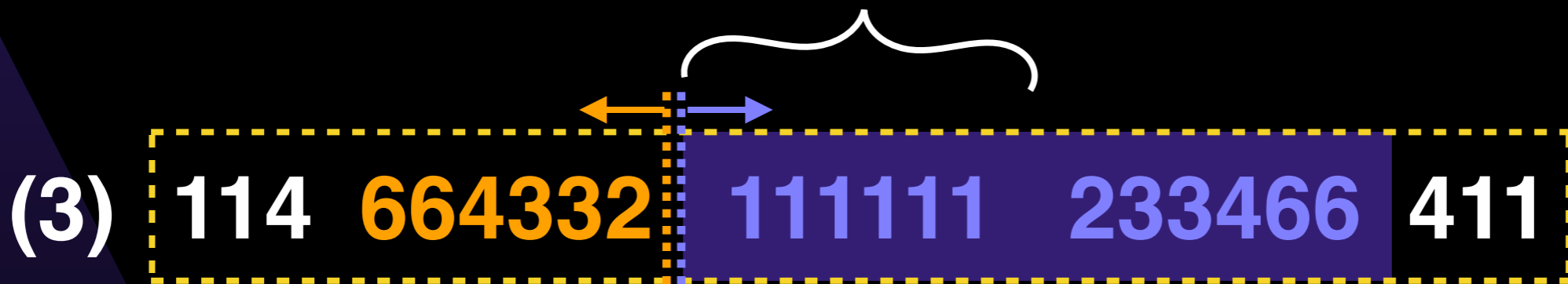


(3)(4)は？

中心はどんな数？



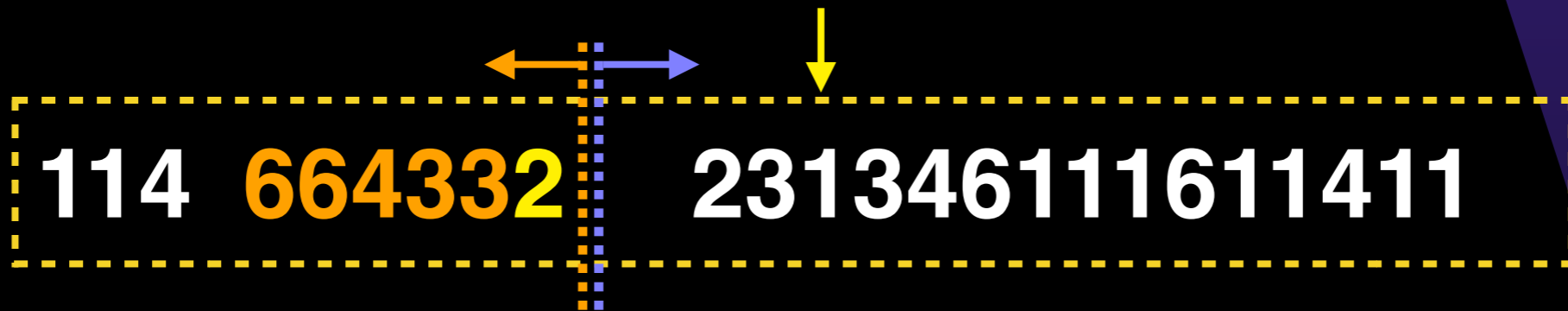
から右に見ていって初めに見つかる, の1つ左より小さい数



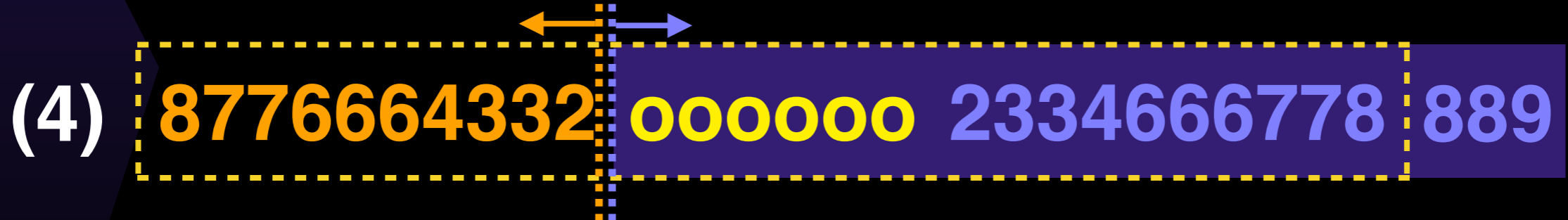
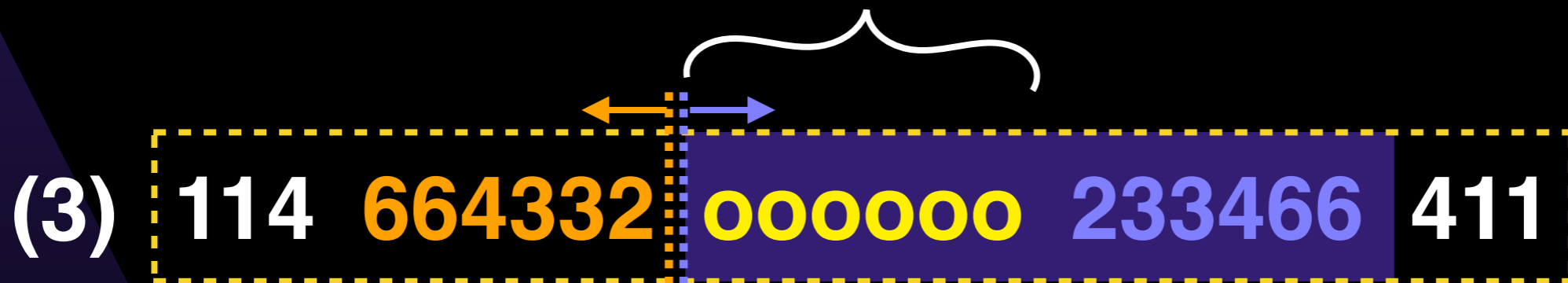


(3)(4)は？

中心をoとおく



から右に見ていって初めに見つかる, の1つ左より小さい数





現在の状況



Solve()をつくるう

- Solve(Lr, o, RI)
- 中心をO(N)個試す ←(1)(2)
 - できるだけ長い回文を見つけ, Lr=左端, RI=右端
 - o = -1 (なし)
- Lr,RIをO(N)通り試す ←(3)(4)
 - o = RI から右に見て行って初めに見つかる, S[Lr] より小さい数
- Solve() を O(N)~O(NlogN) ぐらいで





Solve(): パターン(2)(4)

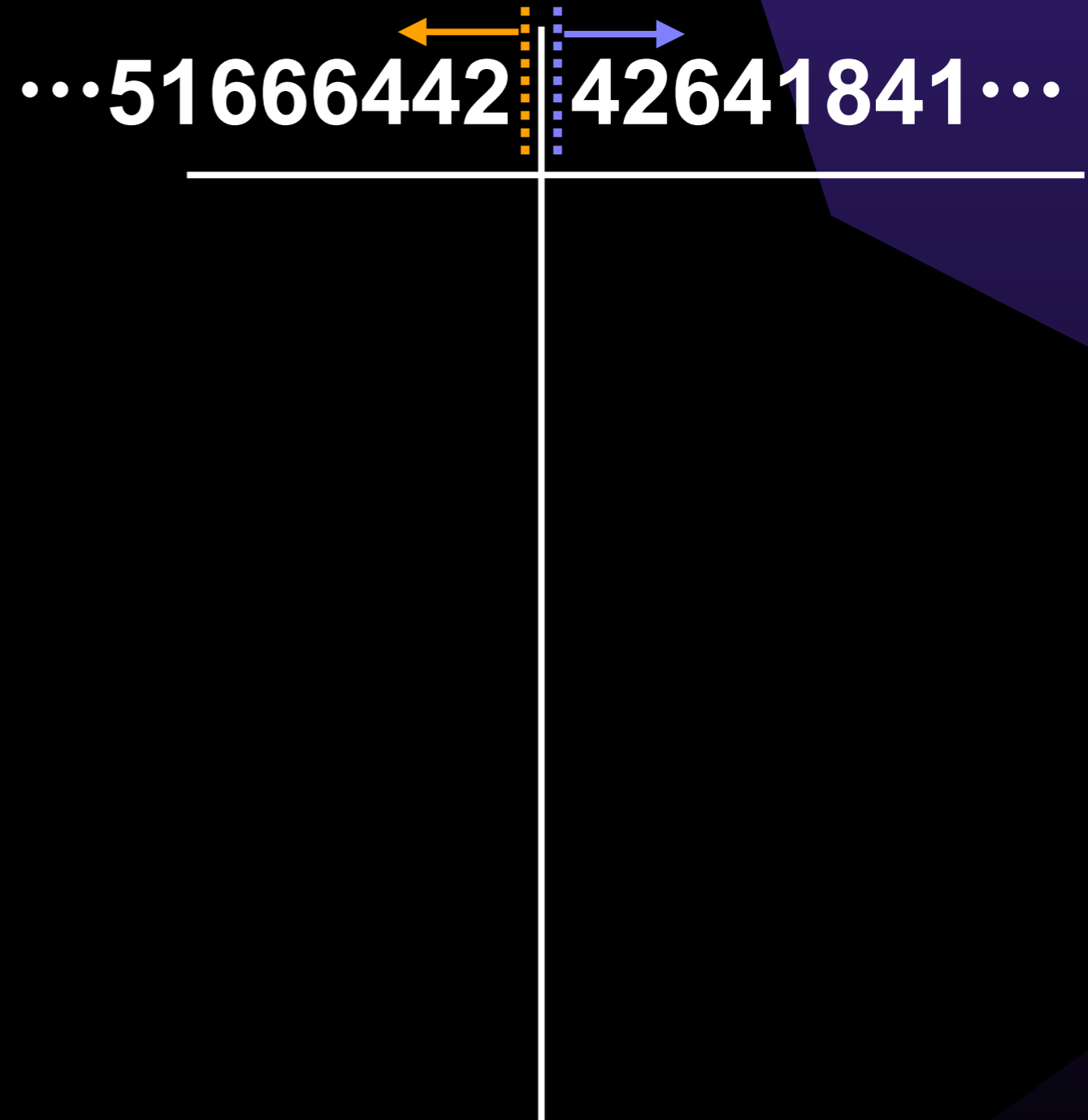
(ソート列が回文をはみ出る)

- (1) 114 664332 514415 233466 411
- (2) 8776664332 514415 2334666778 889
- (3) 114 664332 oooooo 233466 411
- (4) 8776664332 oooooo 2334666778 889



左側の文字を数える

0 = 1

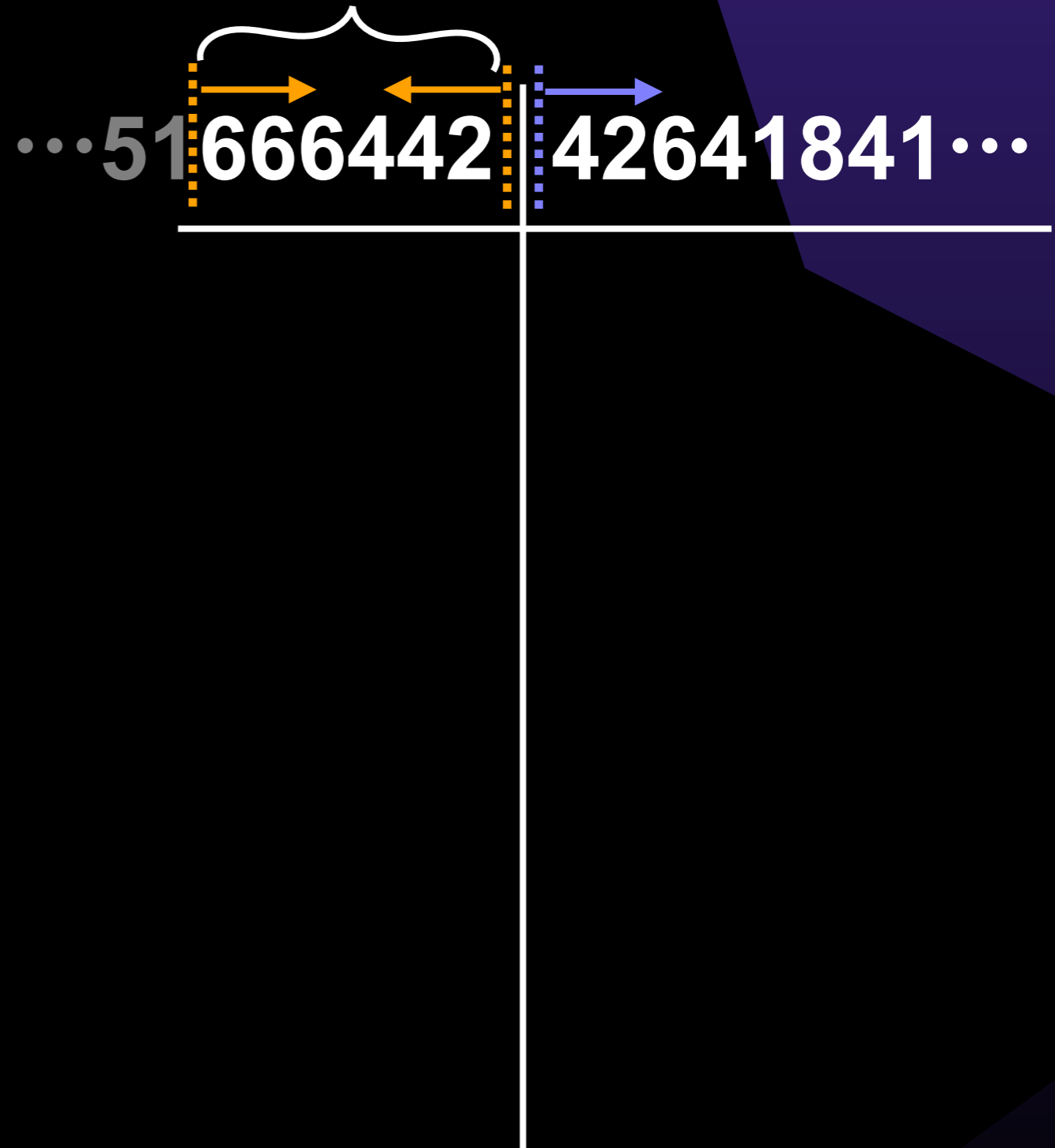




左側の文字を数える

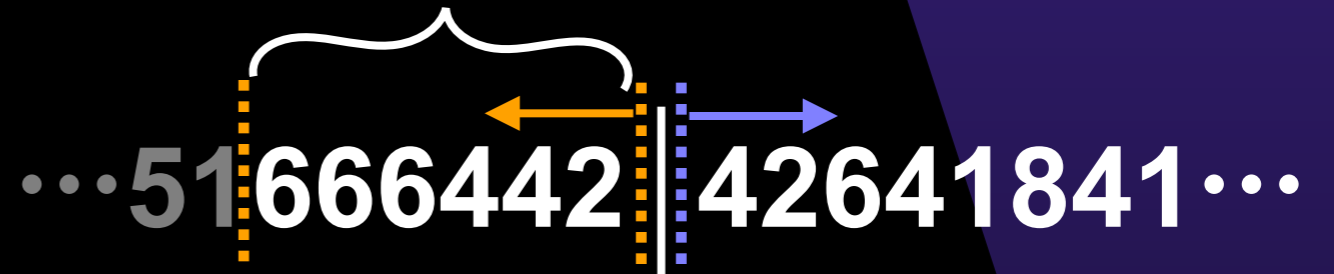
0 = 1

降順の部分の数える

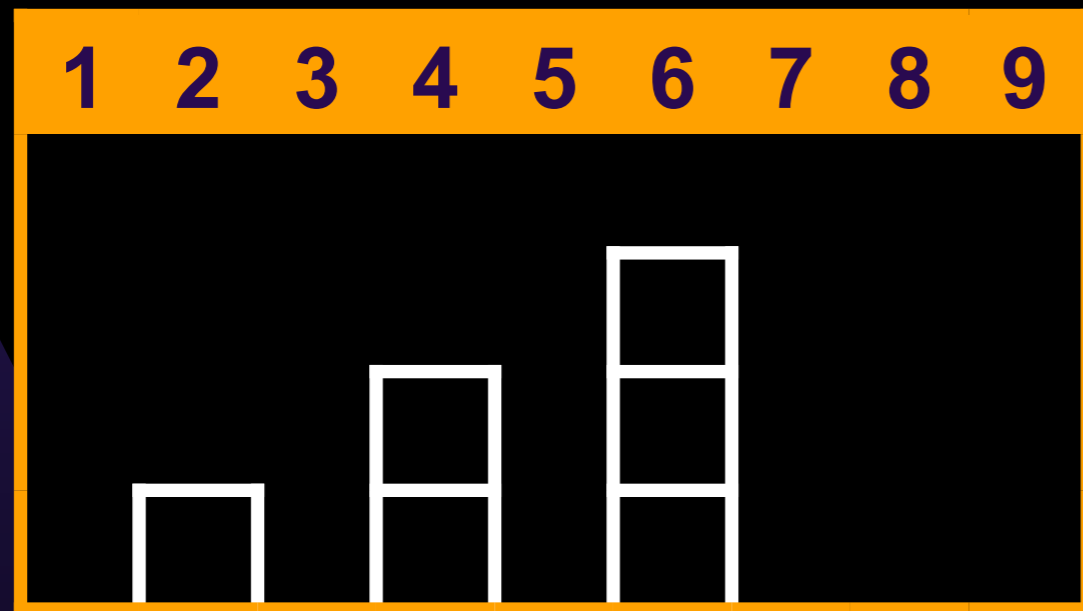


左側の文字を数える

降順の部分の数える



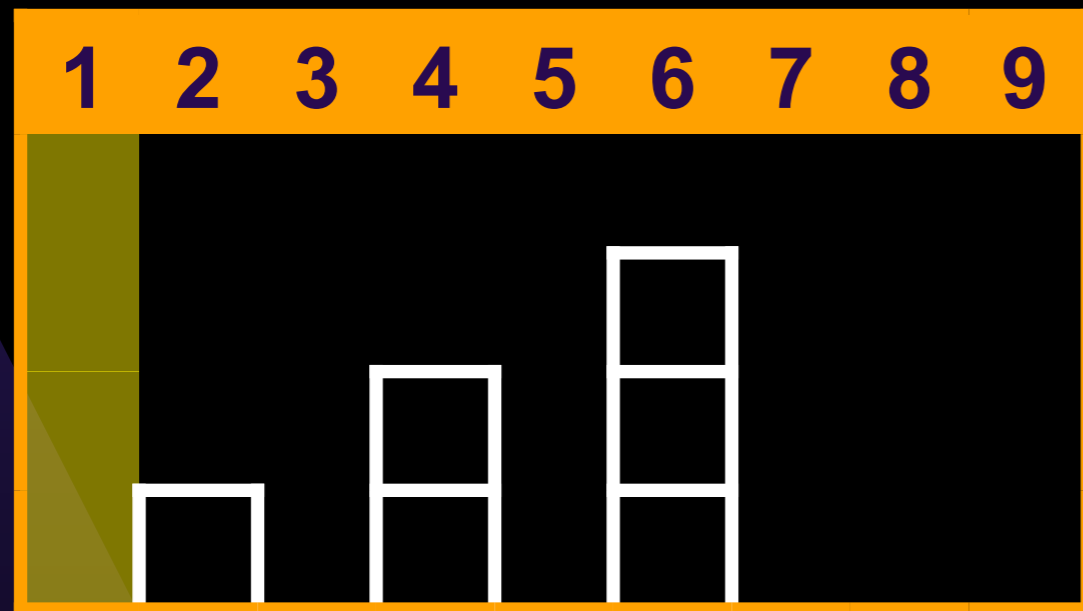
○ = 1





左側の文字を数える

○ = 1



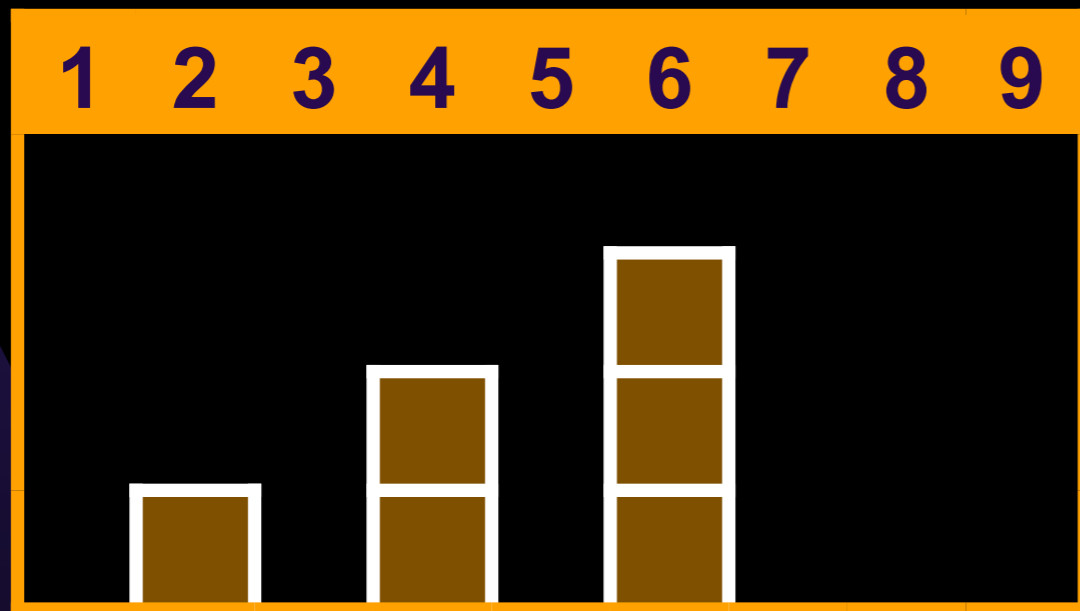
どれだけ入れてもOK

$\cdots 51 \overset{\leftarrow}{|} 666442 \overset{\rightarrow}{|} 42641841 \cdots$



左側の文字を数える

○ = 1



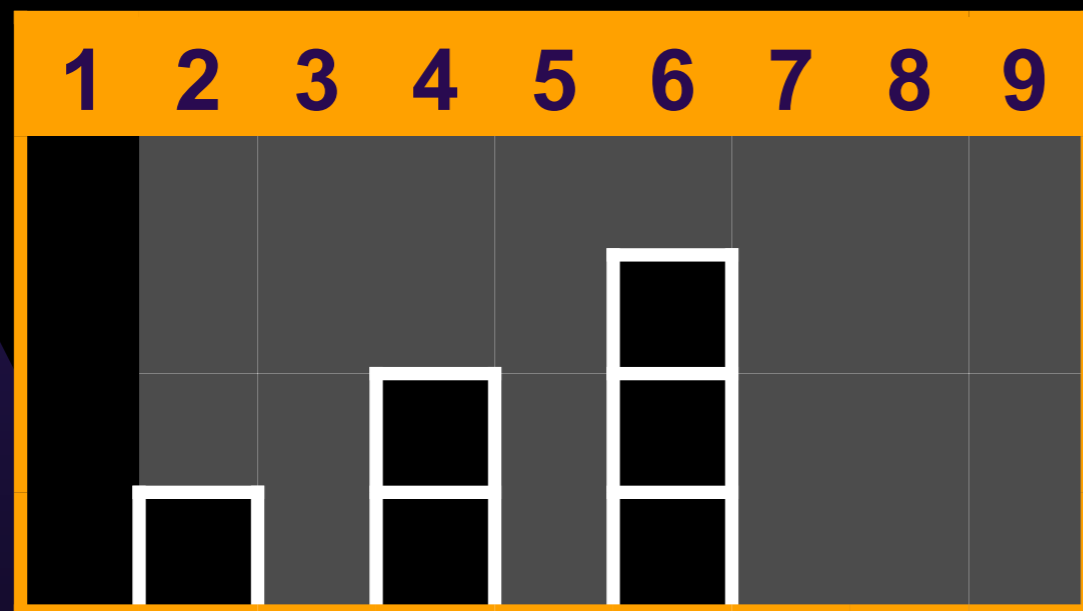
マッチする可能性アリ

$\cdots 51$
666442
42641841
 \cdots



左側の文字を数える

○ = 1



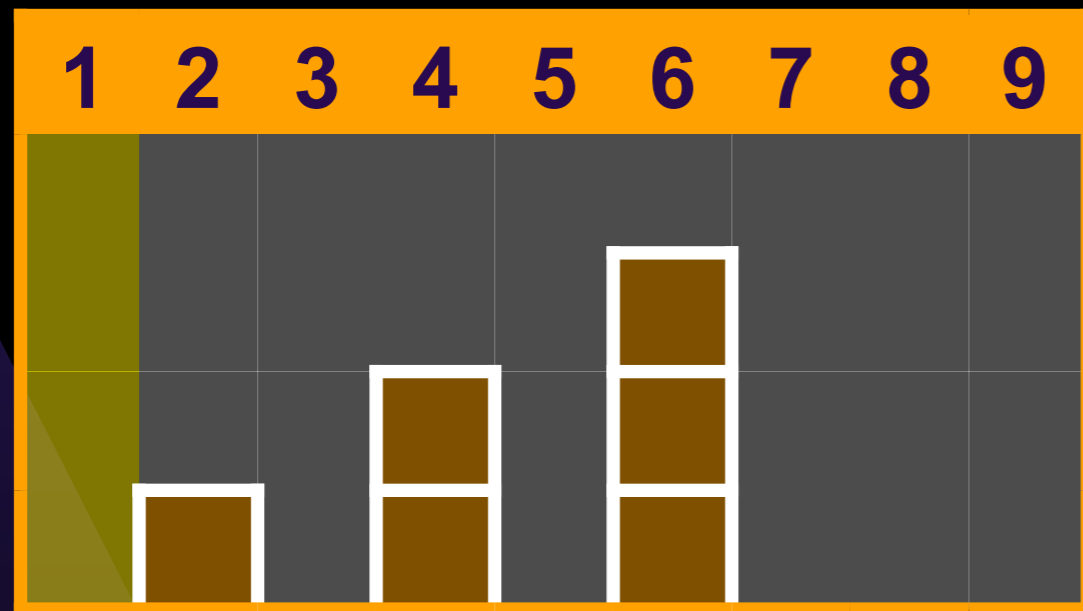
マッチしない

$\cdots 51$
666442
42641841
 \cdots



左側の文字を数える

○ = 1

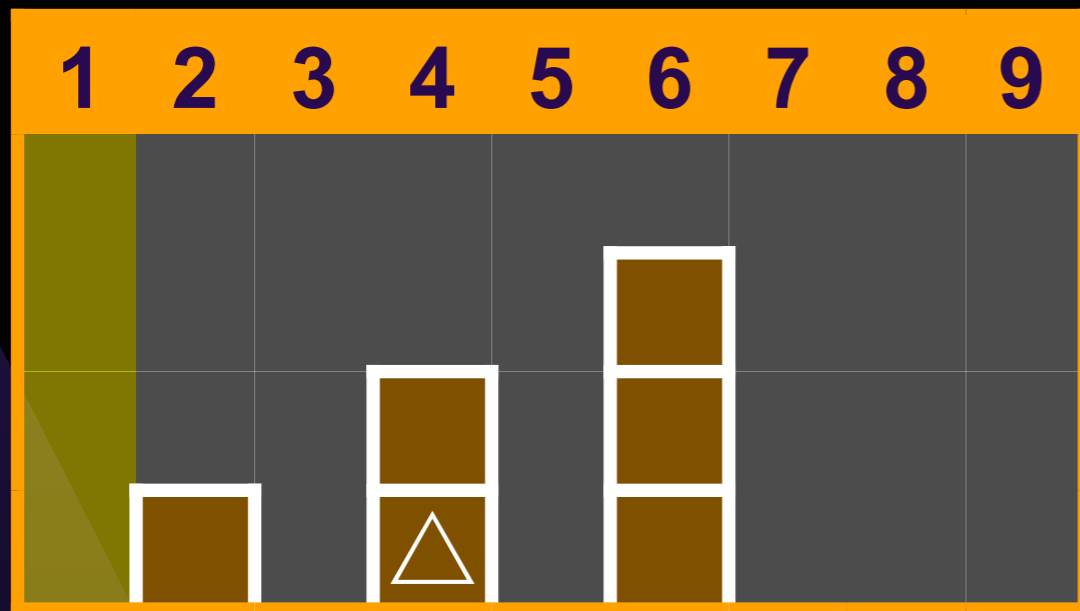


$\cdots 51$
666442
42641841
 \cdots



右側を伸ばしていく：
4はマッチしない

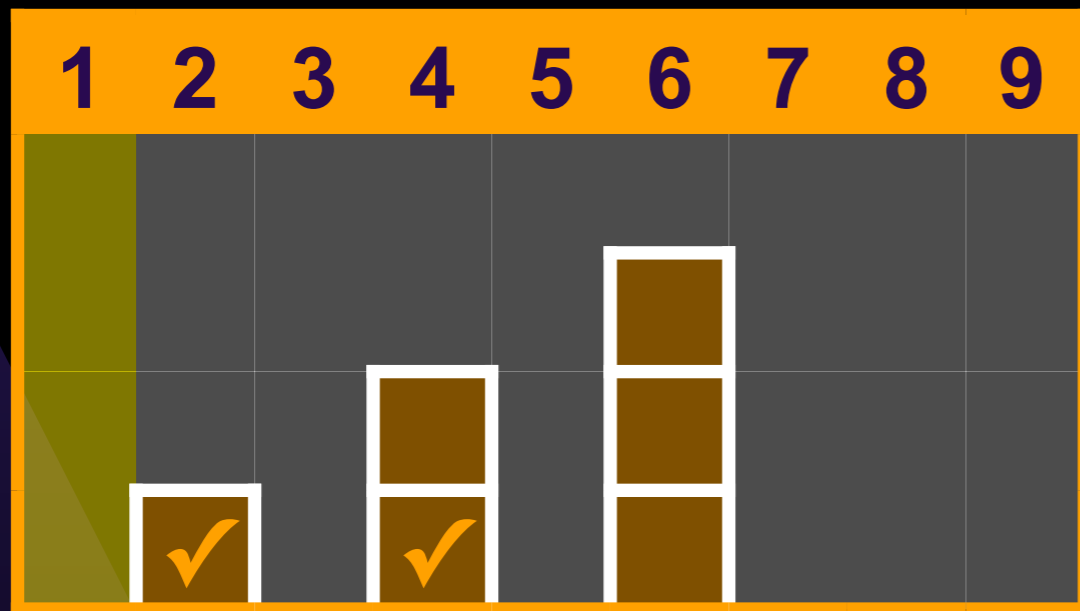
○ = 1





右側を伸ばしていく：
2がマッチして4もマッチする

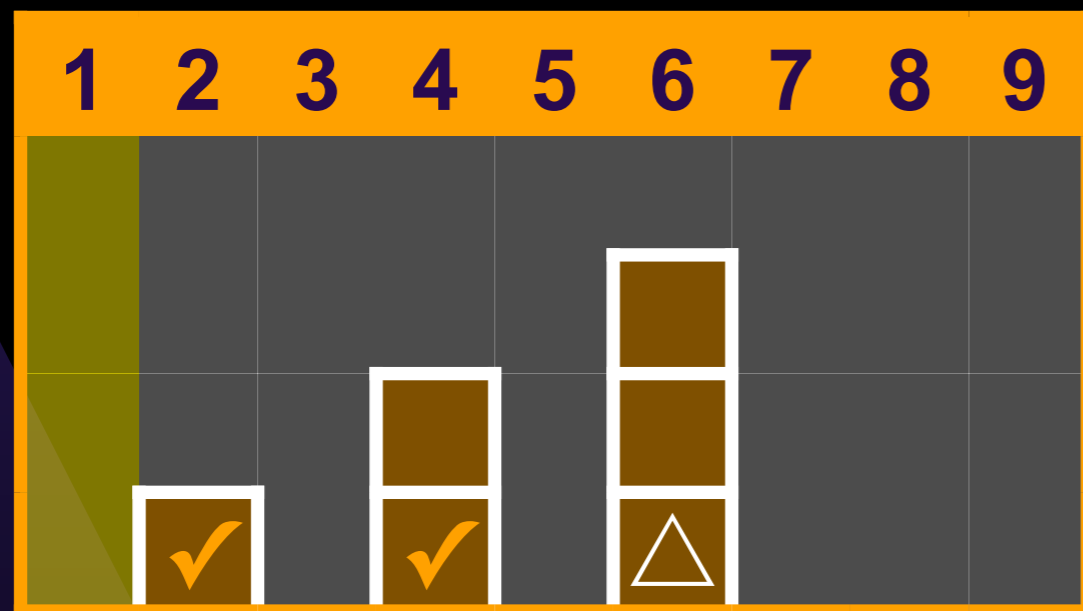
○ = 1





右側を伸ばしていく：
6はマッチしない

○ = 1

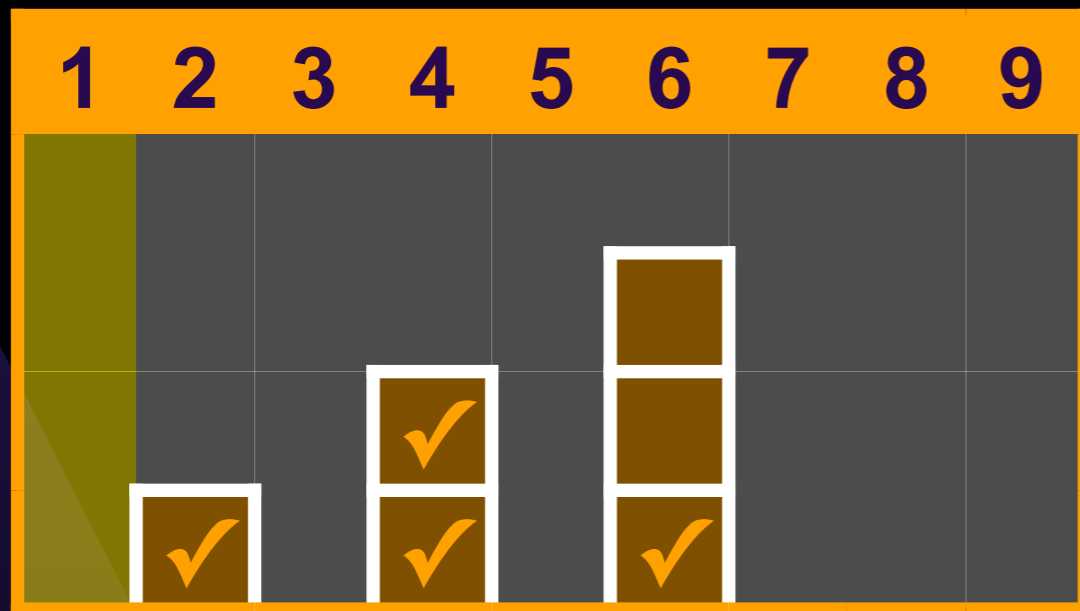


666442	42641841...
666442	4
666442	24
666442	246



右側を伸ばしていく：
4がマッチして6がマッチする

o = 1

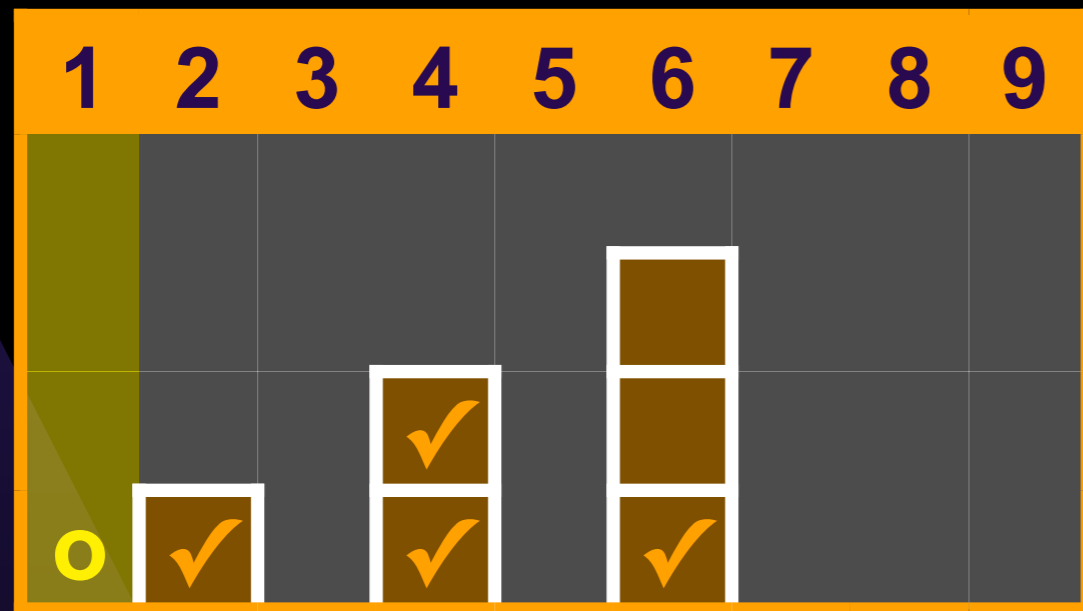


666442	←	42641841...	→
666442		4	
666442		24	
666442		246	
666442		2446	



右側を伸ばしていく：
1は○なので中心が伸びる

○ = 1

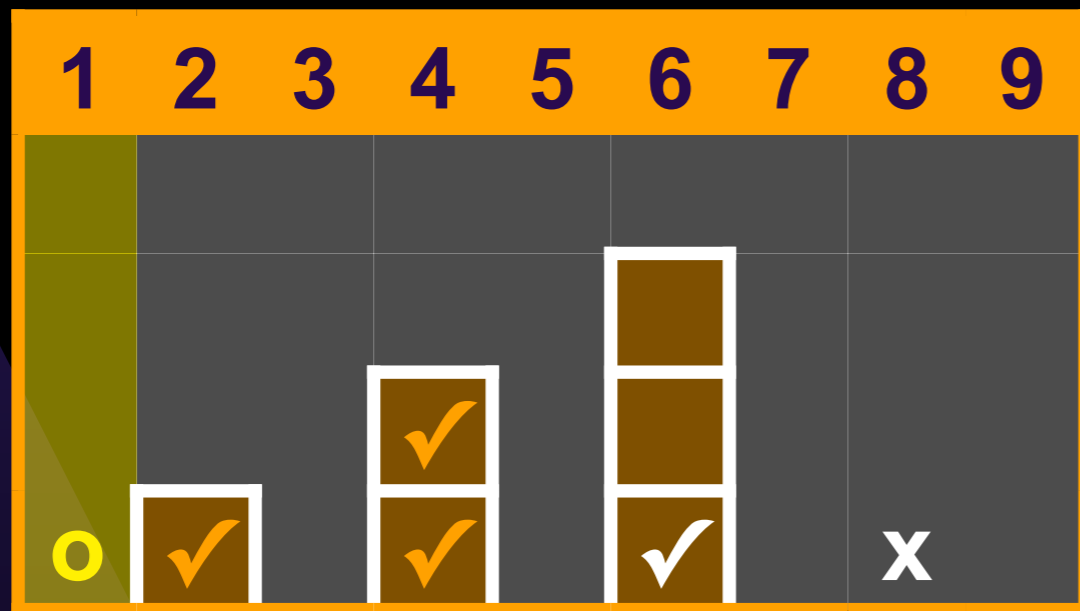


666442	42641841...
666442	4
666442	24
666442	246
666442	2446
666442	12446

右側を伸ばしていく：
8はゴミ



○ = 1

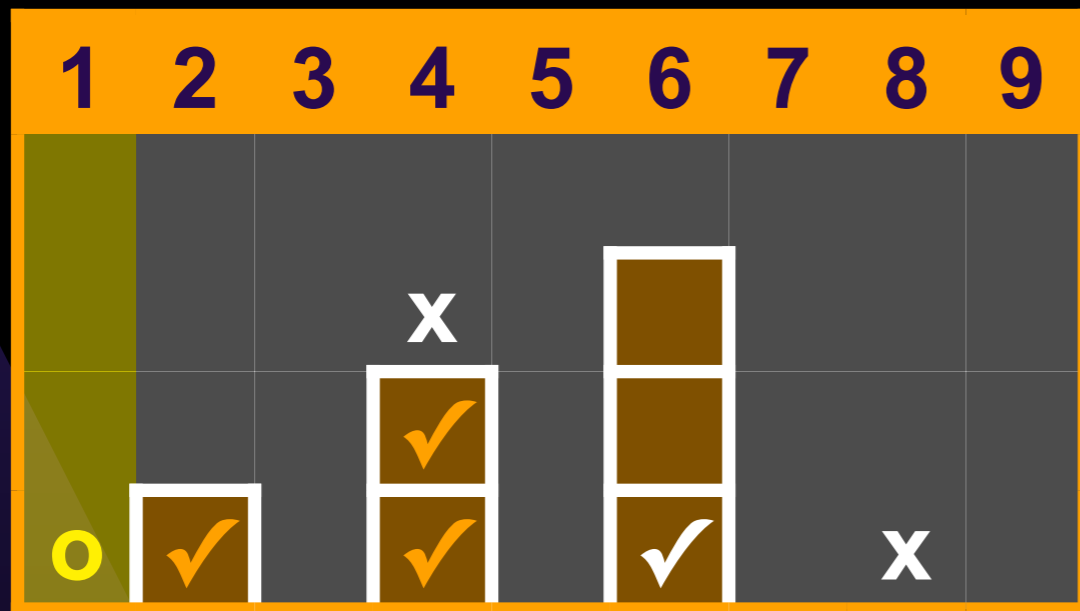


666442	42641841...
666442	4
666442	24
666442	246
666442	2446
666442	12446
666442	124468



右側を伸ばしていく：
4はゴミ，大きい6もゴミに

○ = 1

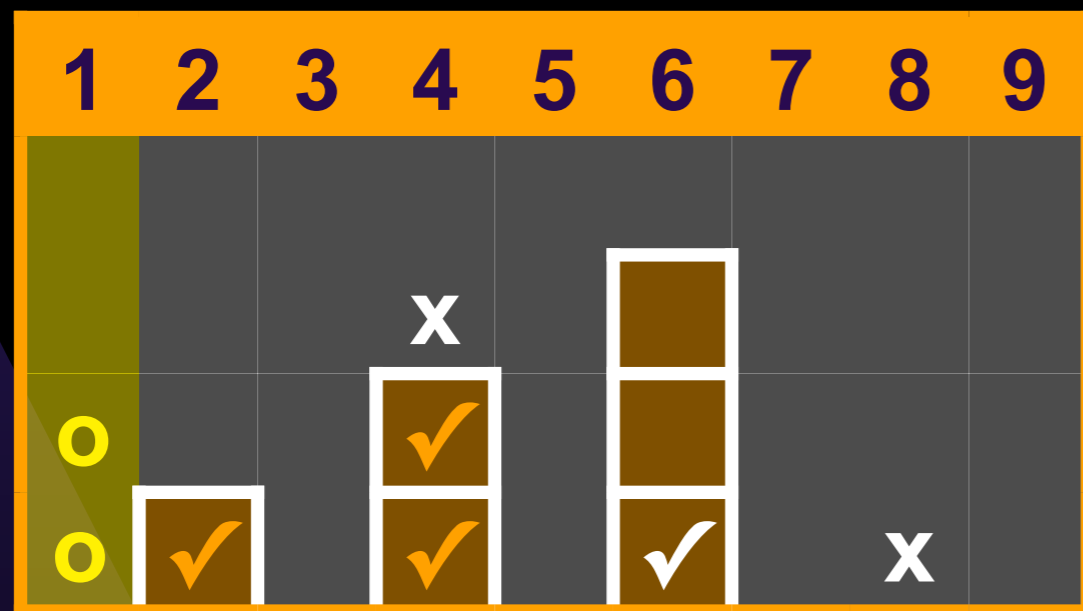


666442	42641841...
666442	4
666442	24
666442	246
666442	2446
666442	12446
666442	124468
666442	1244468



右側を伸ばしていく：
ゴミが入っても○は伸びうる

○ = 1

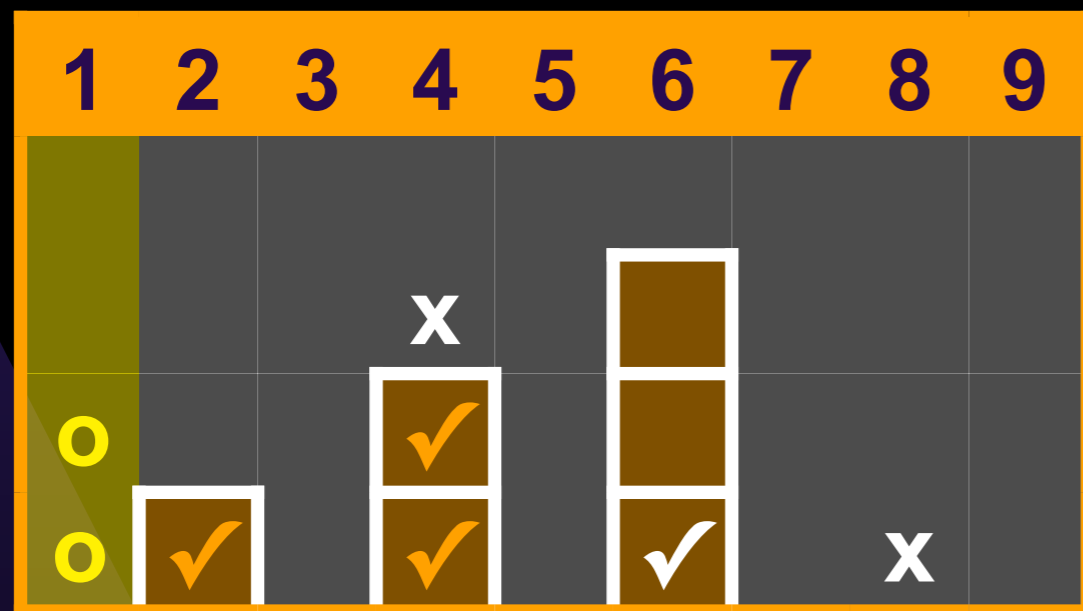


666442	42641841...
666442	4
666442	24
666442	246
666442	2446
666442	12446
666442	124468
666442	1244468
666442	11244468
⋮	⋮



こういうのを カウントする

○ = 1



○ の個数 = 2

✓ の右端までの個数 = 4

x の左端までの個数 = 3

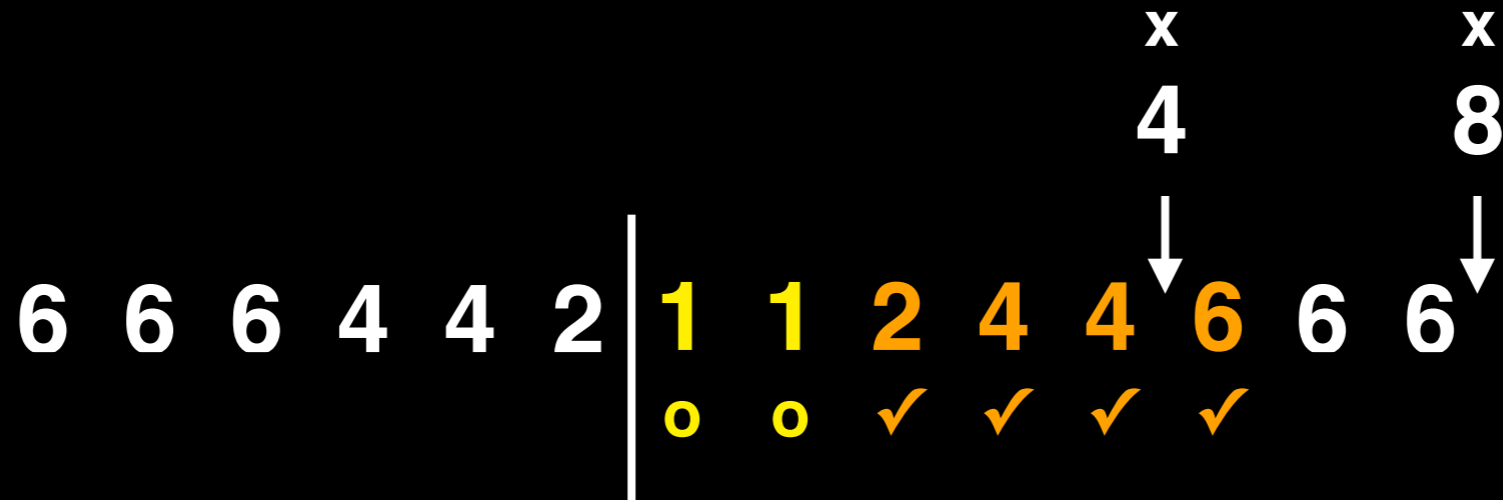
長さ = $\min(4, 3) + 2 + \min(4, 3) = 8$

666442	42641841...
666442	4
666442	24
666442	246
666442	2446
666442	12446
666442	124468
666442	1244468
666442	11244468
⋮	⋮



こういうのを カウントする

66644211244468



o の個数 = 2



✓ の右端までの個数 = 4



x の左端までの個数 = 3

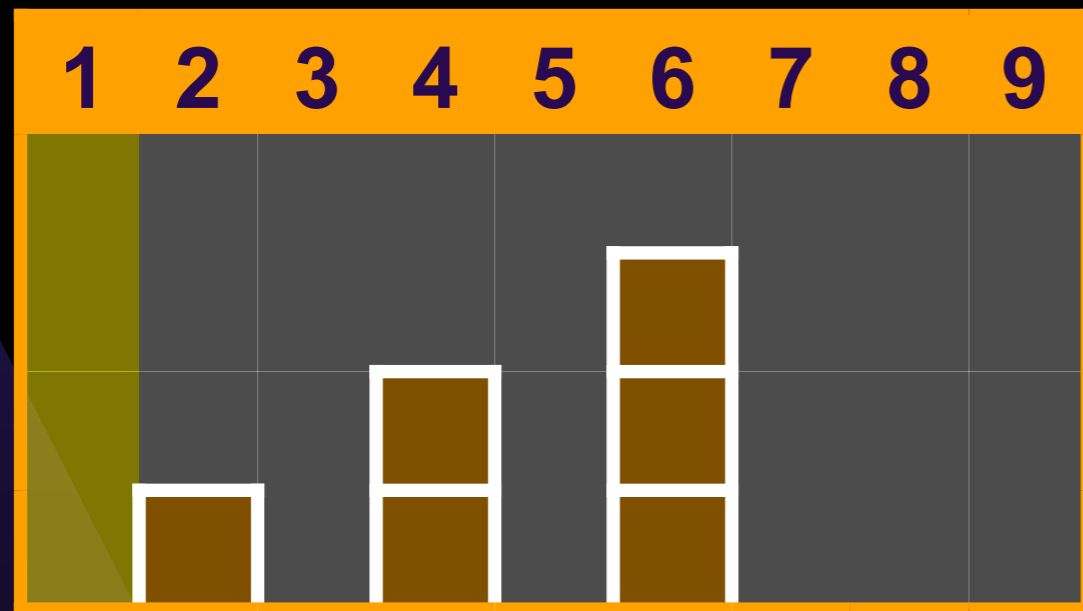


長さ = $\min(4, 3) + 2 + \min(4, 3) = 8$



右側を伸ばしていく：
左側の文字を数える

o = 1



o の個数 = 0

✓ の右端までの個数 = 0

x の左端までの個数 = 6

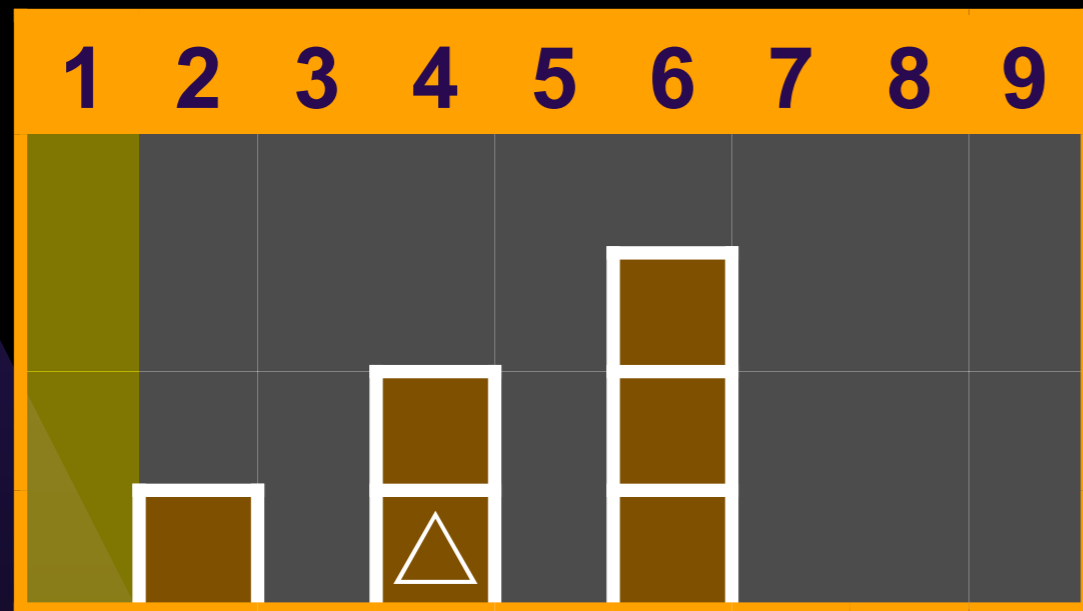
長さ = $\min(0, 6) + 0 + \min(0, 6) = 0$

666442 | 42641841...



右側を伸ばしていく：
4はマッチしない

○ = 1

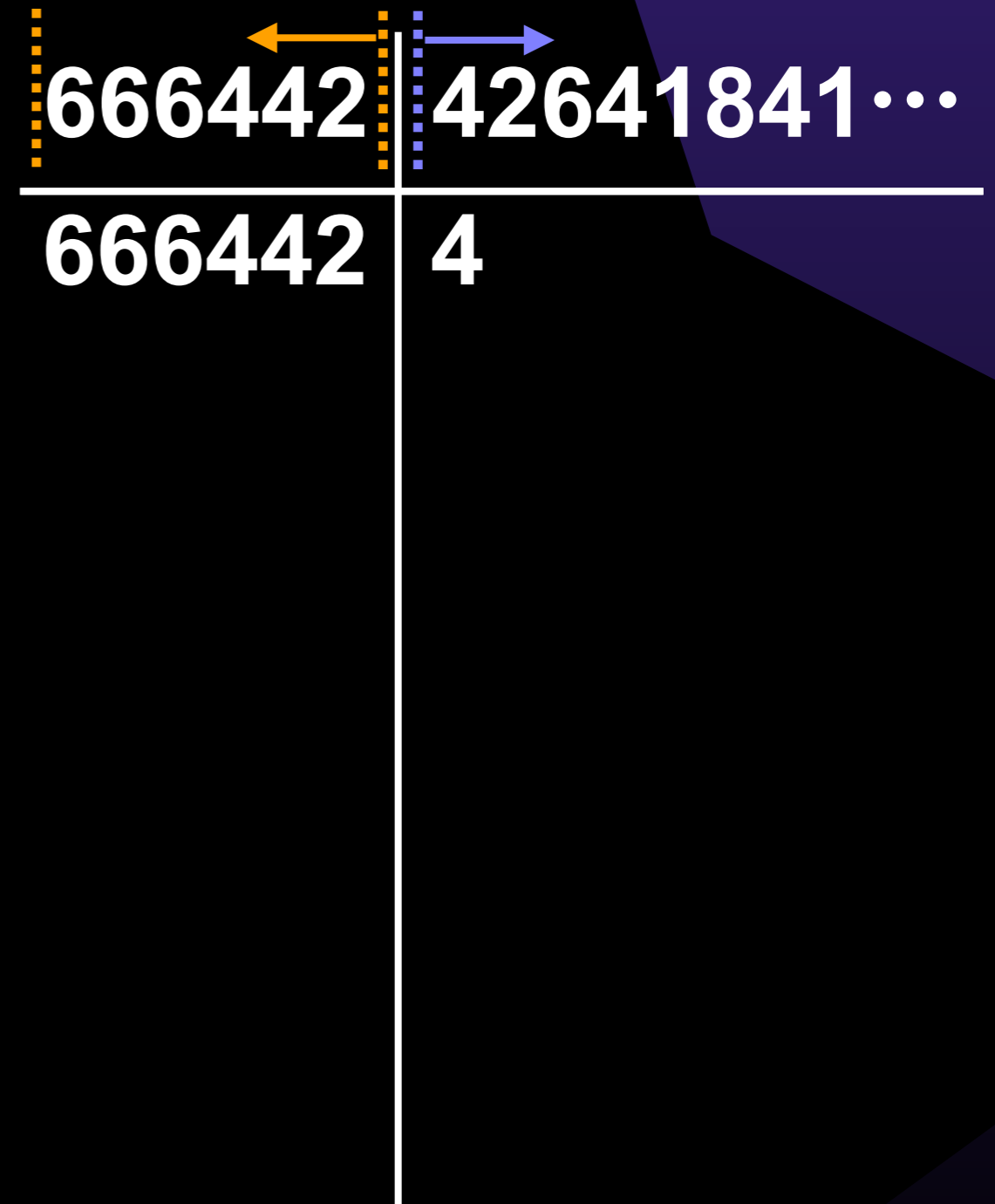


○ の個数 = 0

✓ の右端までの個数 = 0

x の左端までの個数 = 6

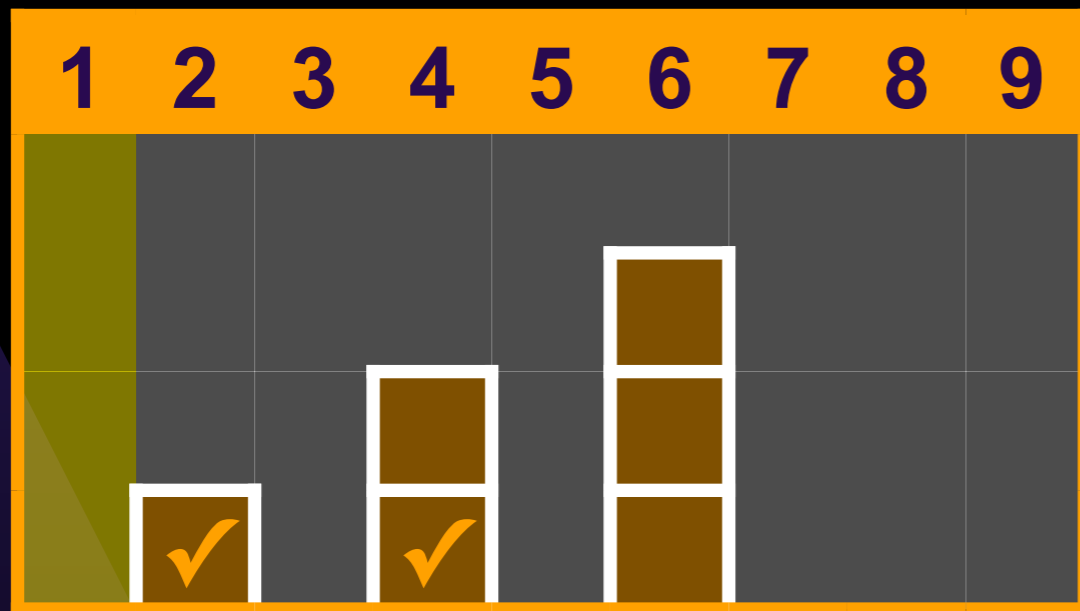
長さ = $\min(0, 6) + 0 + \min(0, 6) = 0$





右側を伸ばしていく：
2がマッチして4もマッチする

o = 1



o の個数 = 0

✓ の右端までの個数 = 2

x の左端までの個数 = 6

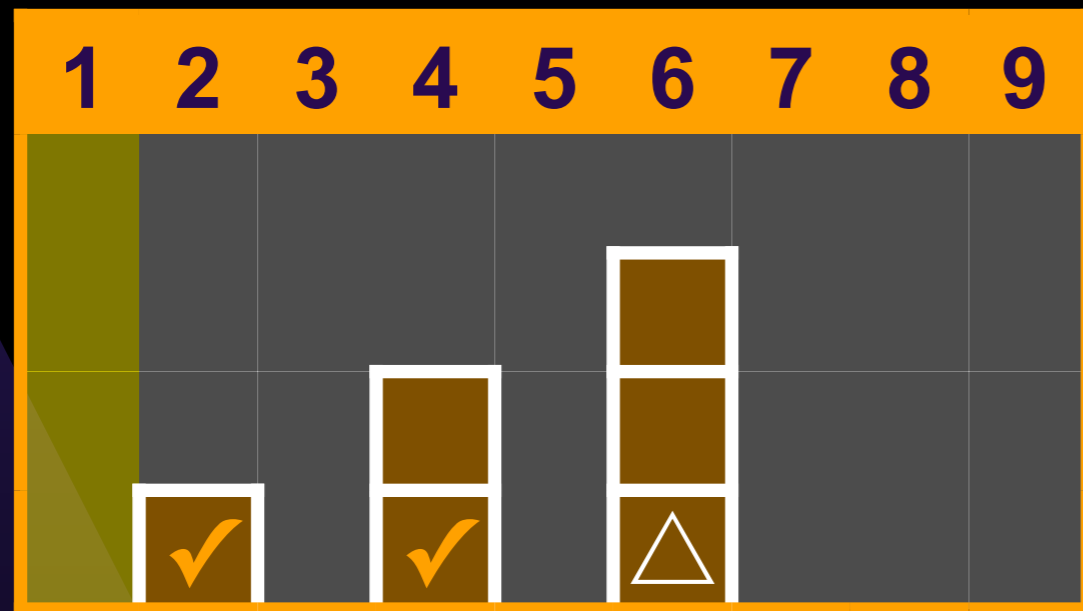
長さ = $\min(2, 6) + 0 + \min(2, 6) = 4$





右側を伸ばしていく：
6はマッチしない

o = 1



o の個数 = 0

✓ の右端までの個数 = 2

x の左端までの個数 = 6

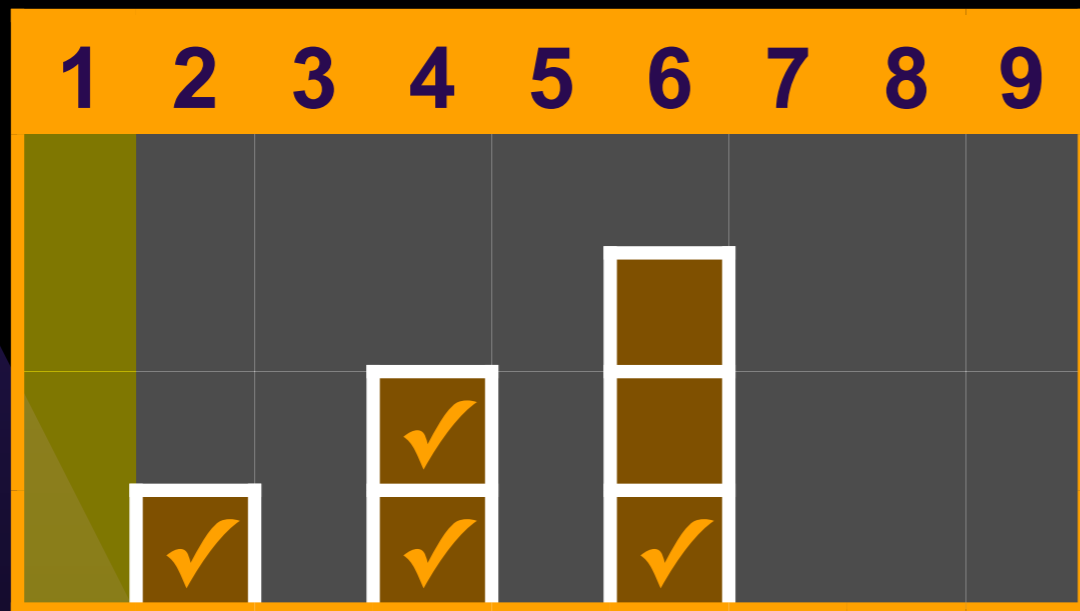
長さ = $\min(2, 6) + 0 + \min(2, 6) = 4$

666442	←	→	42641841...
666442			4
666442			24
666442			246



右側を伸ばしていく：
4がマッチして6がマッチする

o = 1



o の個数 = 0

✓ の右端までの個数 = 4

x の左端までの個数 = 6

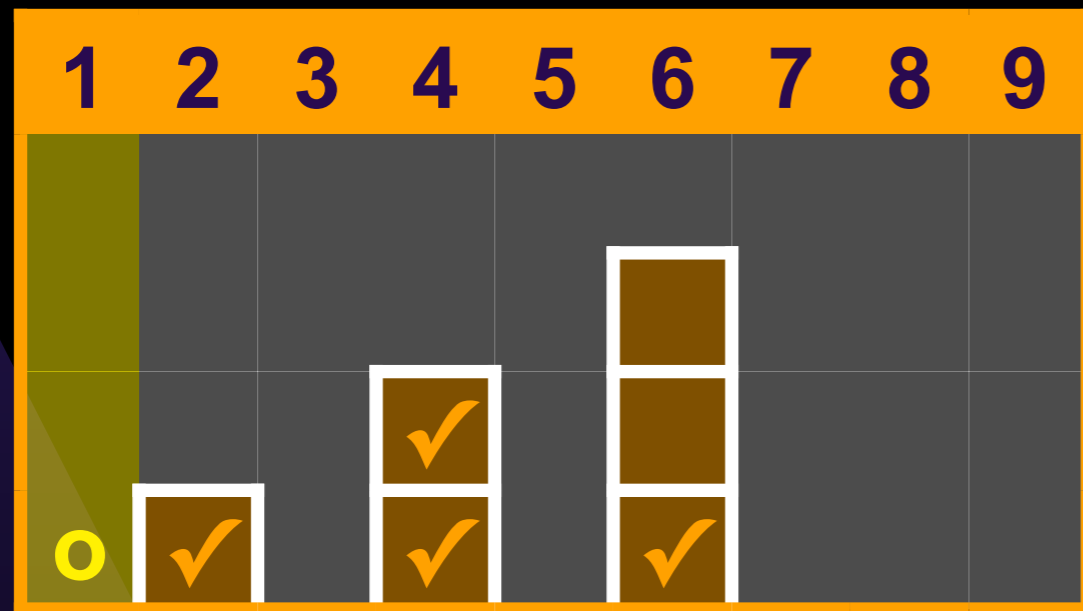
長さ = $\min(4, 6) + 0 + \min(4, 6) = 8$

666442	42641841...
666442	4
666442	24
666442	246
666442	2446



右側を伸ばしていく：
1は○なので中心が伸びる

○ = 1



○ の個数 = 1

✓ の右端までの個数 = 4

x の左端までの個数 = 6

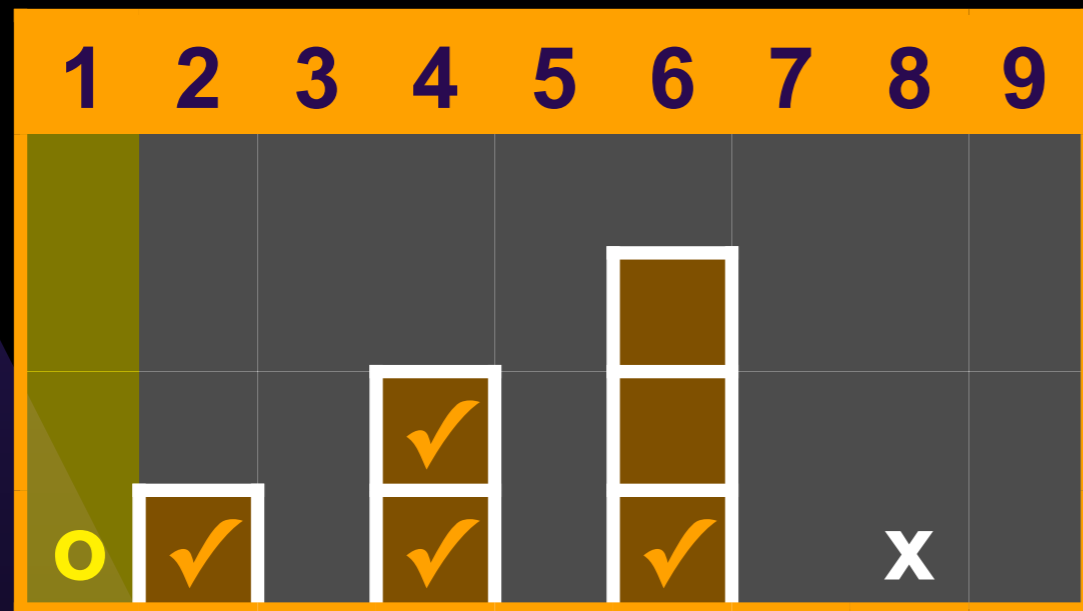
長さ = $\min(4, 6) + 1 + \min(4, 6) = 9$

666442	42641841...
666442	4
666442	24
666442	246
666442	2446
666442	12446



右側を伸ばしていく：
8はゴミ

o = 1



o の個数 = 1

✓ の右端までの個数 = 4

x の左端までの個数 = 6

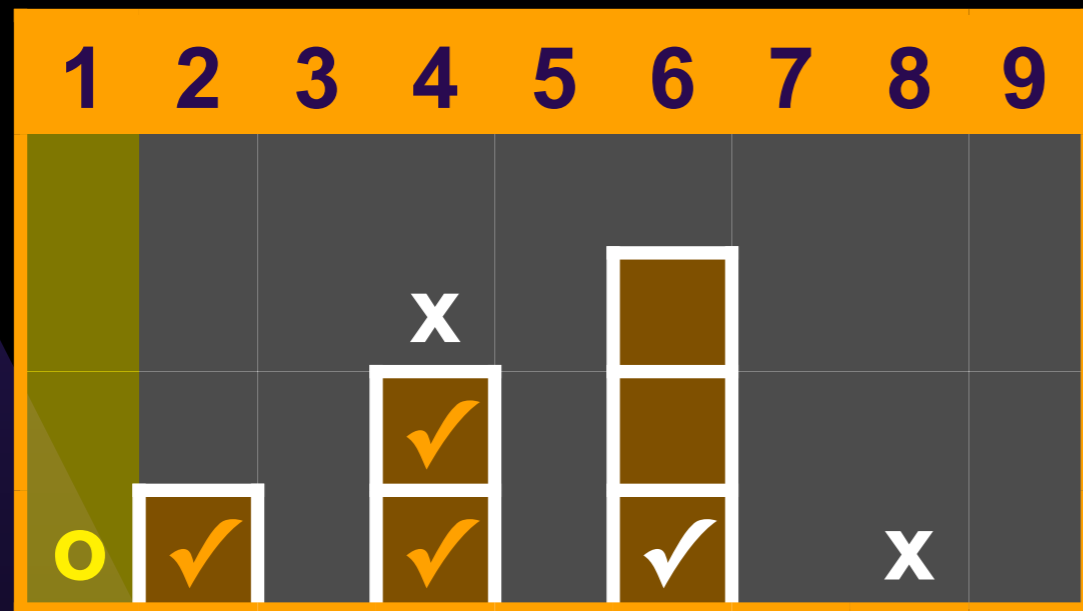
長さ = $\min(4, 6) + 1 + \min(4, 6) = 9$

666442	←	→	42641841...
666442			4
666442			24
666442			246
666442			2446
666442			12446
666442			124468



右側を伸ばしていく：
4はゴミ，大きい6もゴミに

o = 1



o の個数 = 1

✓ の右端までの個数 = 4

x の左端までの個数 = 3

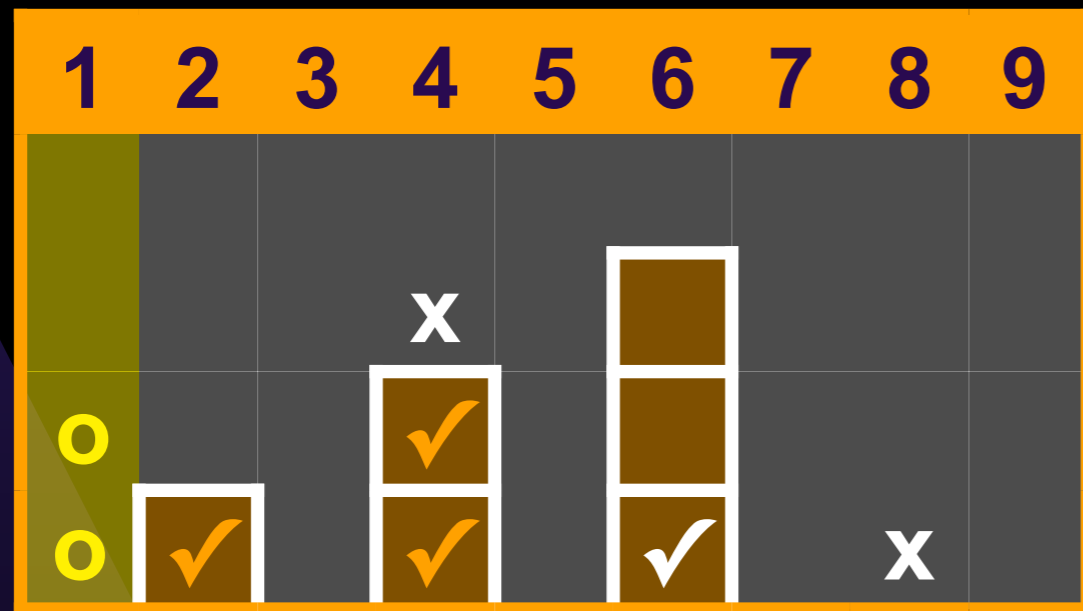
長さ = $\min(4, 3) + 1 + \min(4, 3) = 7$

666442	42641841...
666442	4
666442	24
666442	246
666442	2446
666442	12446
666442	124468
666442	1244468



右側を伸ばしていく：
ゴミが入っても○は伸びうる

○ = 1



○ の個数 = 2

✓ の右端までの個数 = 4

x の左端までの個数 = 3

長さ = $\min(4, 3) + 2 + \min(4, 3) = 8$

666442	42641841...
666442	4
666442	24
666442	246
666442	2446
666442	12446
666442	124468
666442	1244468
666442	11244468
⋮	⋮



Solve(): パターン(1)(3)

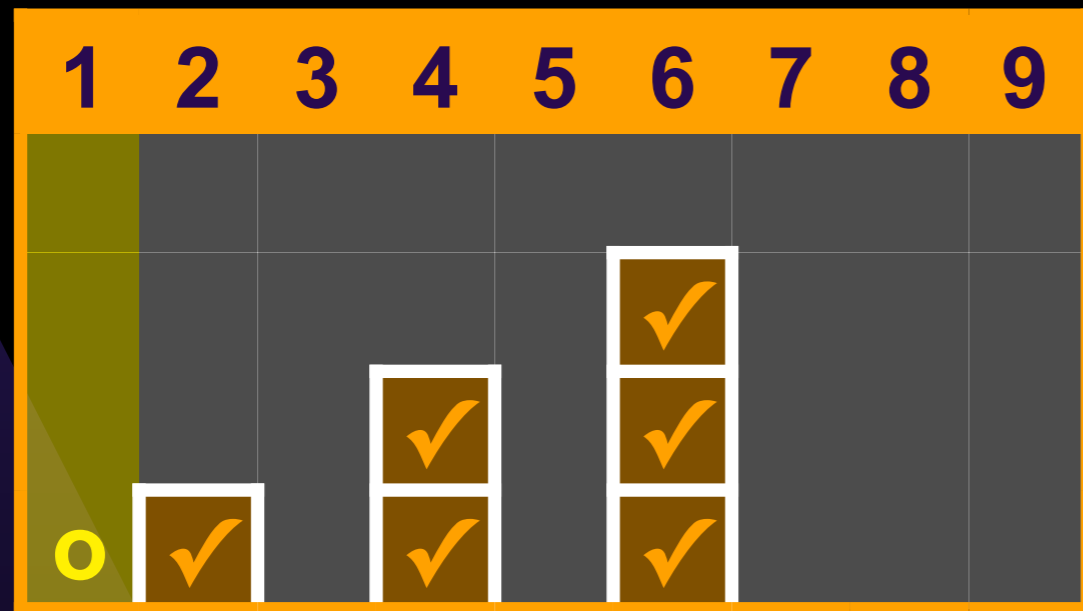
(ソート列が回文をはみ出ない)

- (1) 114 664332 514415 233466 411
- (2) 8776664332 514415 2334666778 889
- (3) 114 664332 000000 233466 411
- (4) 8776664332 000000 2334666778 889



はみ出るパターンと
ほぼ同じだが、

o = 1

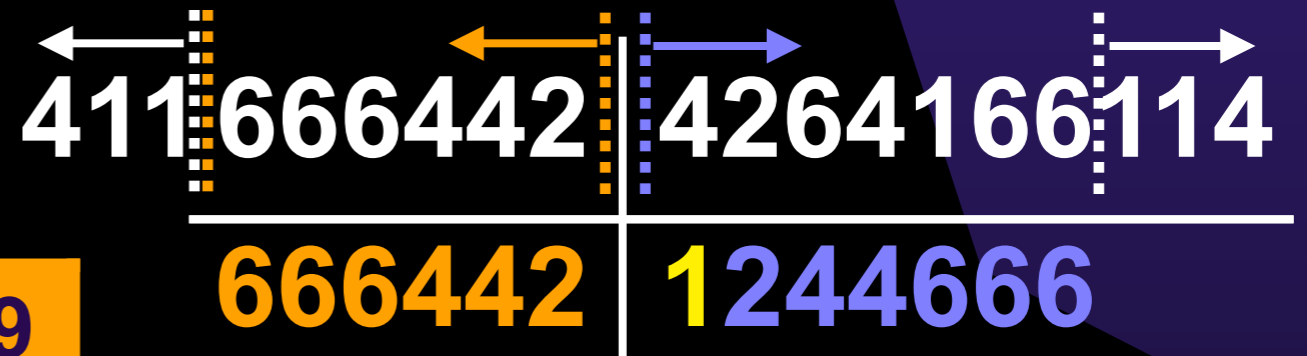


o の個数 = 1

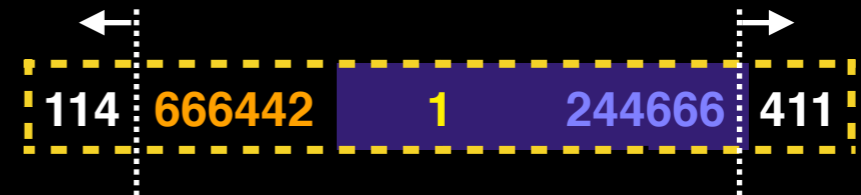
✓ の右端までの個数 = 6

x の左端までの個数 = 6

長さ = $\min(6, 6) + 1 + \min(6, 6) = 13$



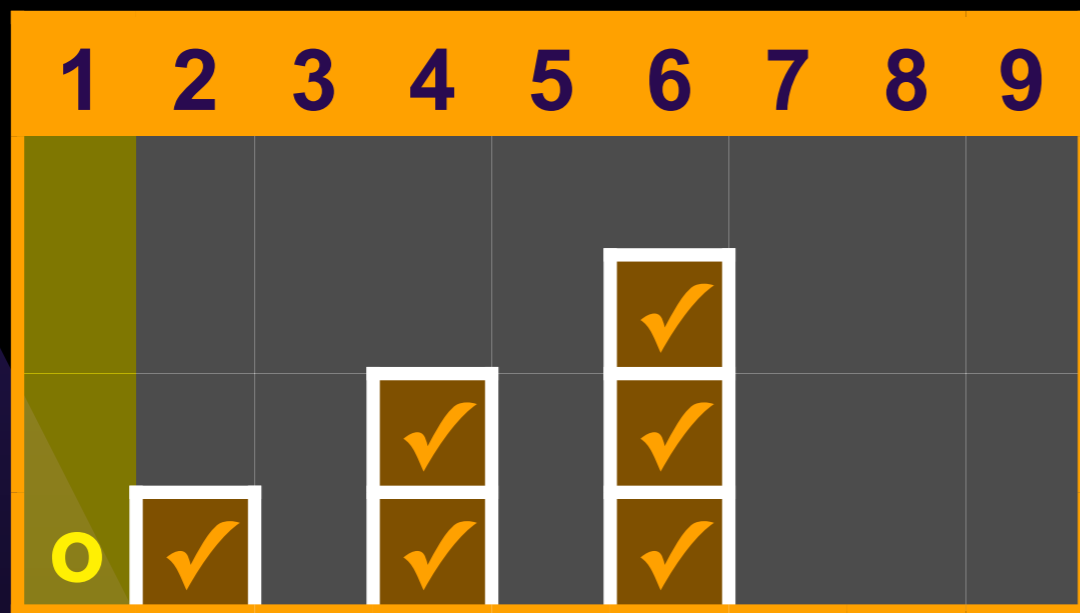
x がないとき, 伸ばせる





はみ出るパターンと
ほぼ同じだが、

o = 1

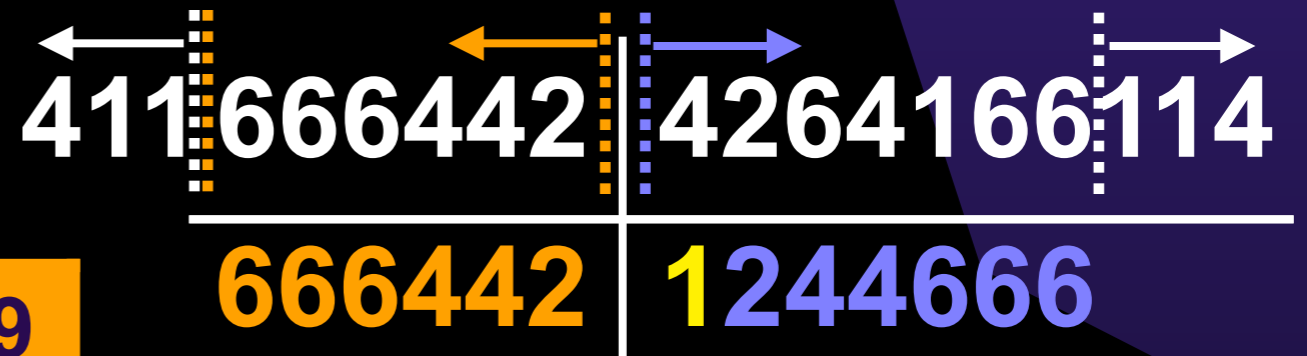


o の個数 = 1

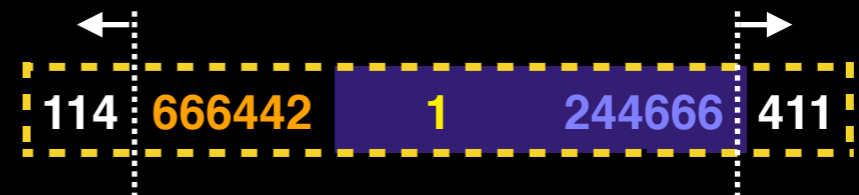
✓ の右端までの個数 = 6

x の左端までの個数 = 6

長さ = $\min(6, 6) + 1 + \min(6, 6) = 13$



x がないとき, 伸ばせる



どれくらい伸ばせるかは,

$DP[\leftarrow \text{の位置}][\rightarrow \text{の位置}]$

として $O(N^2)$ で前計算できる

やったぜ



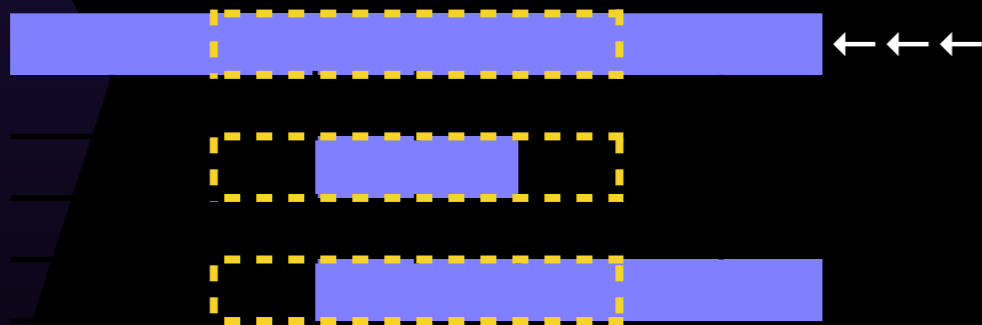
これで本当に
絶対ACするよ！

もしACしなかったら
木の下に埋めて貰っても
構わないよ



全部ソートに注意

バババーン



2124123222

→ × 2121243222

→ ○ 1122222234

解法の概略・まとめ

- 文字列を「反転」させて2回解く
- $DP[a][b]$ = (aまで, と, bから, で一致する文字列の最長)
を $O(N^2)$ で前計算
- 回文の途中までをソートするパターン4つ
 - 初期位置を $O(N)$ 通り
 - 左側はできるだけ長い降順な列をとり,
 - 右側を順に伸ばしていく
 - どちらも $O(N)$, 合計で $O(N^2)$
- 全部ソートするパターンを忘れない



得点分布

Q Z Q Z Q Z Q Z Q Z Q Z Q Z Q Z Q Z Q



得点分布

Q Q Q Q Q Q Q Q Q Q Z Z Z Z Z Z Z Z Z

A = 100

Q = 10

Z = 0