会津合宿 2018 Day3 G 問題
回文部分列 (Palindromic Subsequences)

原案: tsutaj
問題文: tsutaj
解答: tsutaj・kazu・tsukasa_diary
解説: tsutaj

2018 年 9 月 21 日

問題

回文部分列 (Palindromic Subsequences)

- 文字列 S が与えられる
- S の連続とは限らない部分文字列であって回文であるものは何種類？
- 制約
  - $1 \leq |S| \leq 2,000$
  - S は英小文字のみからなる

例

“acpc” に含まれる回文部分列は 5 種類

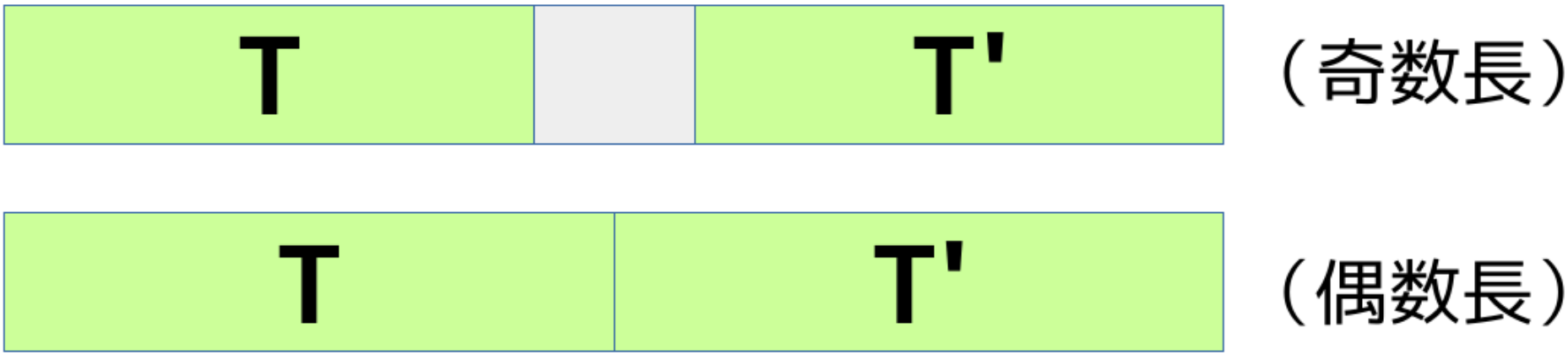
- “a”, “c”, “cc”, “cpc”, “p”
- それぞれの回文は unique に数える

回文の性質

- 回文はどんな性質がある？

回文の性質

- 回文はどんな性質がある？
  - 右から読んでも左から読んでも同じ
  - 文字列の前半を T、T を左右反転させた文字列を T' とおくとき、回文 S は T + (中心の文字) + T' の形をしている

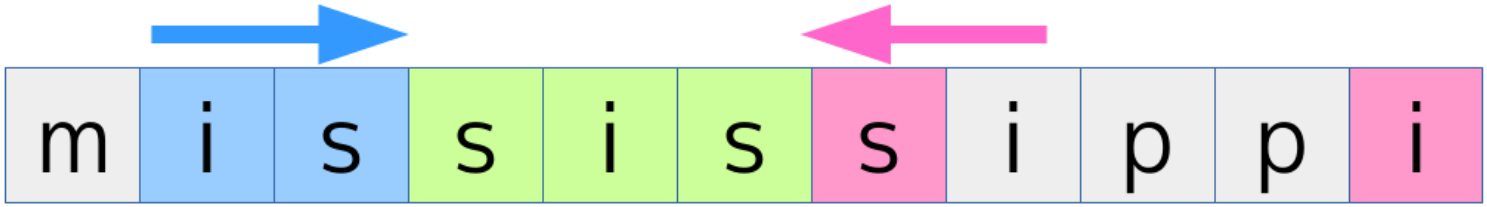


方針

- T としてあり得るものが何種類あるのかを知りたい

方針

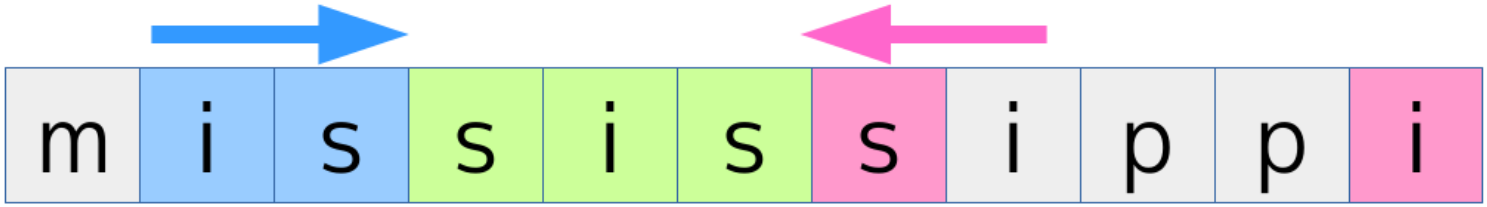
- T としてあり得るものが何種類あるのかを知りたい
- T' は T を反転させたものなので、S を左から走査してできる部分列と、S を右から走査してできる部分列が一致する状況を数え上げたい



青: 左から走査して得た T
赤: 右から走査して得た T'
緑: 奇数長回文のときに中心の文字になりうる候補

方針

- T としてあり得るものが何種類あるのかを知りたい
- T' は T を反転させたものなので、S を左から走査してできる部分列と、S を右から走査してできる部分列が一致する状況を数え上げたい
- T および T' の終端の位置がわかれば、中心の文字としてあり得るものの種類数も数えられる



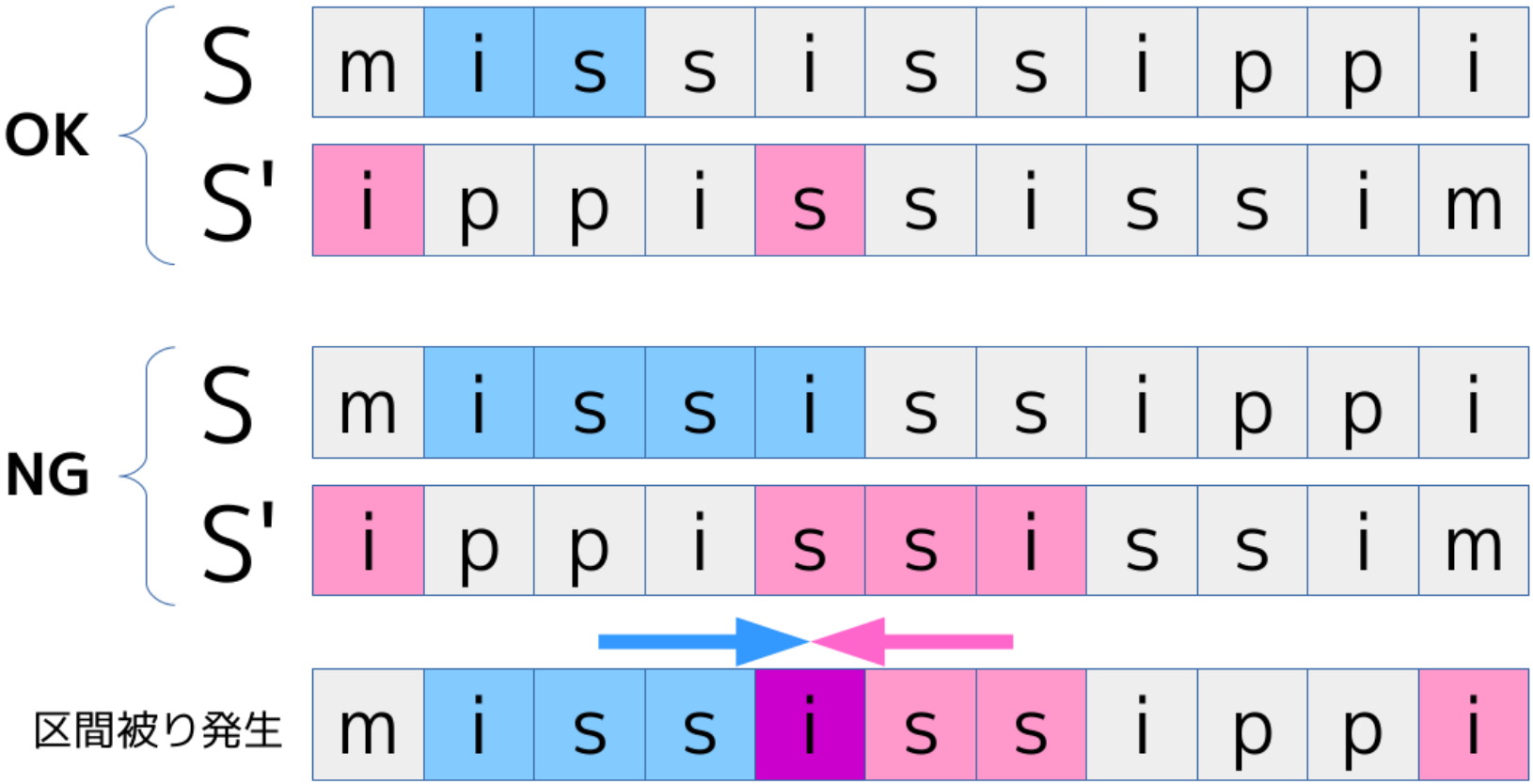
青: 左から走査して得た T
赤: 右から走査して得た T'
緑: 奇数長回文のときに中心の文字になりうる候補

方針

- S の他に、S を反転させた文字列 S' も用意

方針

- S の他に、S を反転させた文字列 S' も用意
- T および T' を求めることは、S と S' の共通部分列を求めることと同じ！
  - ただし区間被りに注意すること



共通部分列

- 2 つの文字列に対する共通部分列の種類数はどのように数え上げる？

共通部分列

- 2 つの文字列に対する共通部分列の種類数はどのように数え上げる？
- DP をしよう！
  - 説明の都合上、2 つの文字列を X, Y と置く

共通部分列

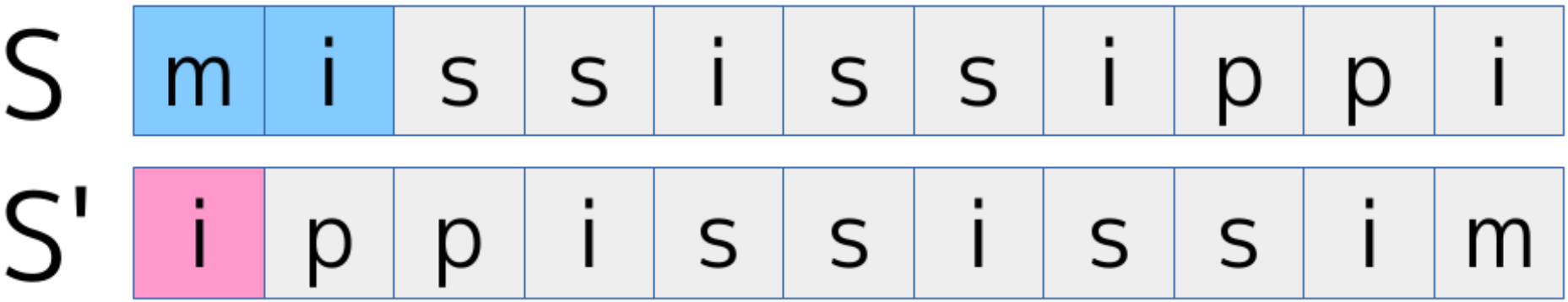
- 2 つの文字列に対する共通部分列の種類数はどのように数え上げる？
- DP をしよう！
  - 説明の都合上、2 つの文字列を X, Y と置く
  - $\mathrm{dp}[i][j] := X$ の i 文字目、Y の j 文字目まで使ってできる共通部分列の種類数
  - 状態 $\mathrm{dp}[i][j]$ で考慮している共通部分列の末尾にアルファベット c を追加するときは、X の i 文字目より後の c の中で最も近いもの・Y の j 文字目より後の c の中で最も近いものを選んで遷移すれば良い

共通部分列

- 2 つの文字列に対する共通部分列の種類数はどのように数え上げる？
- DP をしよう！
  - 説明の都合上、2 つの文字列を X, Y と置く
  - $\mathrm{dp}[i][j] := X$ の i 文字目、Y の j 文字目まで使ってできる共通部分列の種類数
  - 状態 $\mathrm{dp}[i][j]$ で考慮している共通部分列の末尾にアルファベット c を追加するときは、X の i 文字目より後の c の中で最も近いもの・Y の j 文字目より後の c の中で最も近いものを選んで遷移すれば良い
  - 次ページで詳しく解説

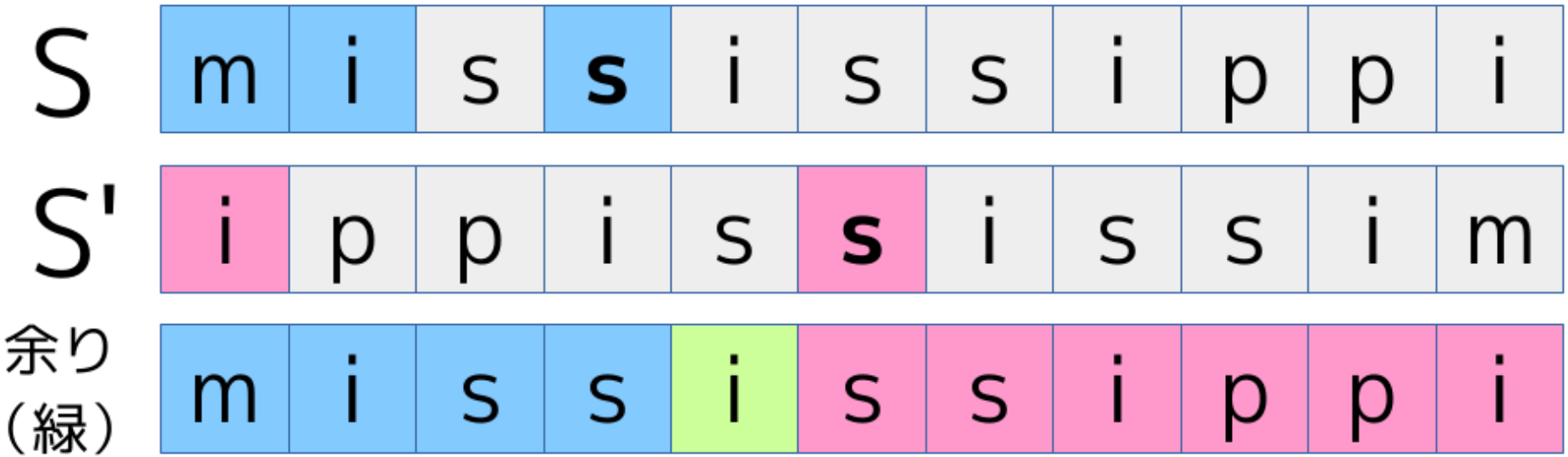
共通部分列

- dp[2][1] の状態から、共通部分列を 1 文字増やしたい
- 's' を増やしたい場合、どの s を持ってくればよいか？



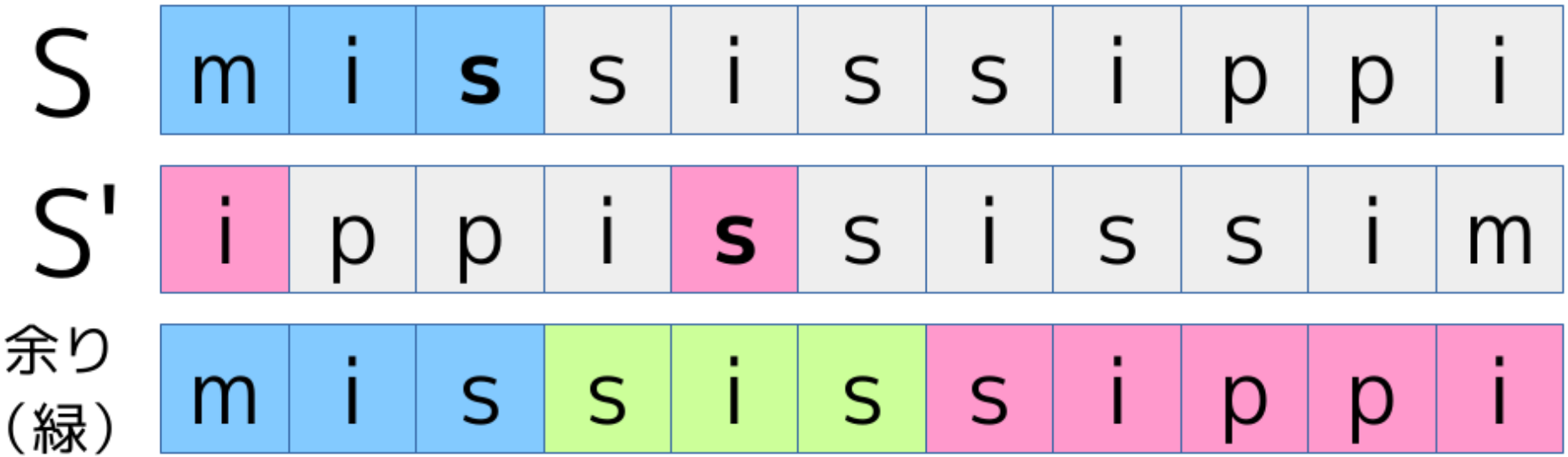
共通部分列

- こういうとり方もあり得るが・・・



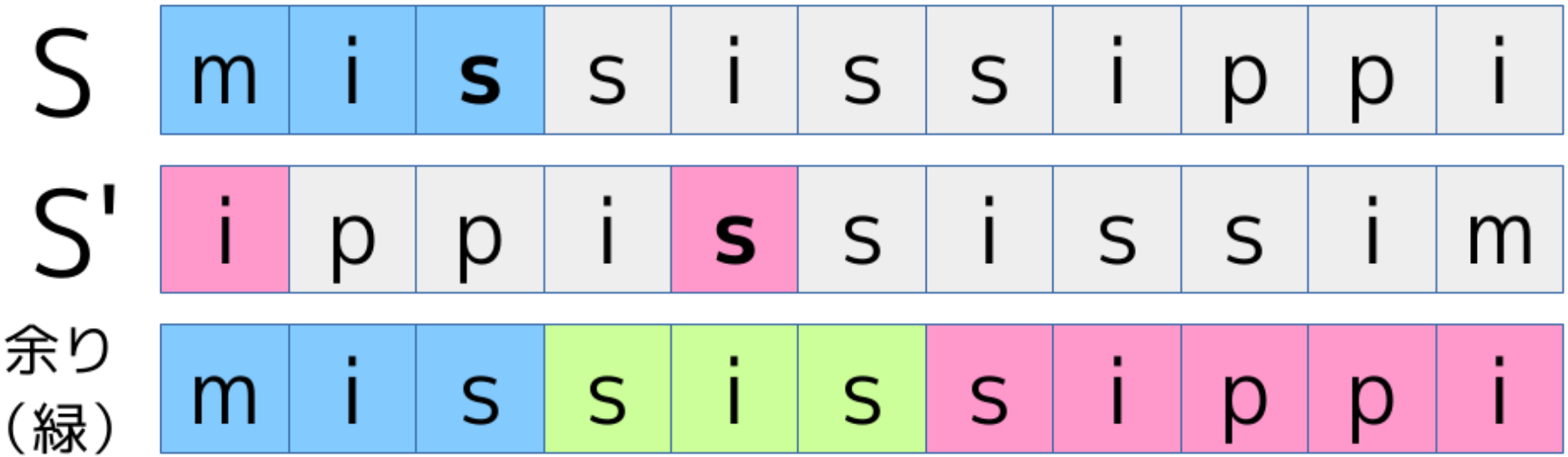
共通部分列

- この方が余りが多くて余裕がある！
  - 余りが多いと、より長い共通部分列を作れる可能性がある



共通部分列

- この方が余りが多くて余裕がある！
  - 余りが多いと、より長い共通部分列を作れる可能性がある
- なので、文字を増やす際は近いものを貪欲に取り、余りを増やしたほうが良い
- この DP によって、共通部分列を unique に数え上げることができる



解法の流れ

回文部分列の数え上げは以下のように解ける

解法の流れ

回文部分列の数え上げは以下のように解ける

- 先ほどの共通部分列の DP をやる

解法の流れ

回文部分列の数え上げは以下のように解ける

- 先ほどの共通部分列の DP をやる
- $\mathrm{dp}[i][j]$ の中で区間として被らないものについて、$\mathrm{dp}[i][j] \times$ (余った区間内にあるアルファベットの種類数 $+1$) を計算して答えに足し合わせる

解法の流れ

回文部分列の数え上げは以下のように解ける

- 先ほどの共通部分列の DP をやる
- $\mathrm{dp}[i][j]$ の中で区間として被らないものについて、$\mathrm{dp}[i][j] \times$ (余った区間内にあるアルファベットの種類数 $+1$) を計算して答えに足し合わせる
  - 余った区間内にあるアルファベットの種類数を掛けるのは、奇数長の回文を数え上げるため
    - 間に 1 文字挟むと奇数長回文になる

解法の流れ

回文部分列の数え上げは以下のように解ける

- 先ほどの共通部分列の DP をやる
- $\mathrm{dp}[i][j]$ の中で区間として被らないものについて、$\mathrm{dp}[i][j]\times$ (余った区間内にあるアルファベットの種類数 $+1$) を計算して答えに足し合わせる
  - 余った区間内にあるアルファベットの種類数を掛けるのは、奇数長の回文を数え上げるため
    - 間に 1 文字挟むと奇数長回文になる
  - $+1$ するのは、偶数長の回文を数え上げるため
    - 間になにも挟まないと偶数長回文になる

解法の流れ

回文部分列の数え上げは以下のように解ける

- 先ほどの共通部分列の DP をやる
- $\mathrm{dp}[i][j]$ の中で区間として被らないものについて、$\mathrm{dp}[i][j]\times$ (余った区間内にあるアルファベットの種類数 $+1$) を計算して答えに足し合わせる
  - 余った区間内にあるアルファベットの種類数を掛けるのは、奇数長の回文を数え上げるため
    - 間に 1 文字挟むと奇数長回文になる
  - $+1$ するのは、偶数長の回文を数え上げるため
    - 間になにも挟まないと偶数長回文になる
  - 空文字列を誤って答えに足さないよう注意！
    - 最後に -1 するか、例外処理するか、お好みで

解法の流れ

回文部分列の数え上げは以下のように解ける

- 先ほどの共通部分列の DP をやる
- $\mathrm{dp}[i][j]$ の中で区間として被らないものについて、$\mathrm{dp}[i][j]\times$ (余った区間内にあるアルファベットの種類数 $+1$) を計算して答えに足し合わせる
  - 余った区間内にあるアルファベットの種類数を掛けるのは、奇数長の回文を数え上げるため
    - 間に 1 文字挟むと奇数長回文になる
  - $+1$ するのは、偶数長の回文を数え上げるため
    - 間になにも挟まないと偶数長回文になる
  - 空文字列を誤って答えに足さないよう注意！
    - 最後に -1 するか、例外処理するか、お好みで
- 足しあわせた結果が答え

計算量

この解法の計算量は？

計算量

この解法の計算量は？

- 共通部分列 DP は「インデックス i より後に登場するアルファベット c の中で最も近いものはどれか」を前処理することで $O(|S|^2)$
  - 前処理は線形でも可能だし二乗でも可能
  - どちらを書いても構いません

計算量

この解法の計算量は？

- 共通部分列 DP は「インデックス i より後に登場するアルファベット c の中で最も近いものはどれか」を前処理することで $O(|S|^2)$
  - 前処理は線形でも可能だし二乗でも可能
  - どちらを書いても構いません
- 答えを足し合わせる時は、「指定区間内にアルファベットは何種類あるか」を前処理することで $O(|S|^2)$
  - 各アルファベットについて累積和を用意すれば、何種類あるかは $O(1)$
  - 足し合わせるときに参照する場所の候補が $O(|S|^2)$ なので、この処理は全体で $O(|S|^2)$

計算量

この解法の計算量は？

- 共通部分列 DP は「インデックス i より後に登場するアルファベット c の中で最も近いものはどれか」を前処理することで $O(|S|^2)$
  - 前処理は線形でも可能だし二乗でも可能
  - どちらを書いても構いません
- 答えを足し合わせる時は、「指定区間内にアルファベットは何種類あるか」を前処理することで $O(|S|^2)$
  - 各アルファベットについて累積和を用意すれば、何種類あるかは $O(1)$
  - 足し合わせるときに参照する場所の候補が $O(|S|^2)$ なので、この処理は全体で $O(|S|^2)$
- したがって問題全体で $O(|S|^2)$ で解ける！

Writer 解・統計

- Writer 解
  - tsutaj (C++・78 行・1985 bytes)
  - kazu (C++・83 行・2329 bytes)
  - tsukasa_diary (C++・60 行・1422 bytes)
- 統計
  - AC / tried: 11 / 15 (73.3 %)
  - First AC
    - On-site: acpc_aizulim (79 min)
    - On-line: LLma (23 min)