JOI 2021/2022
壊れた機器 2
(Broken Device 2)

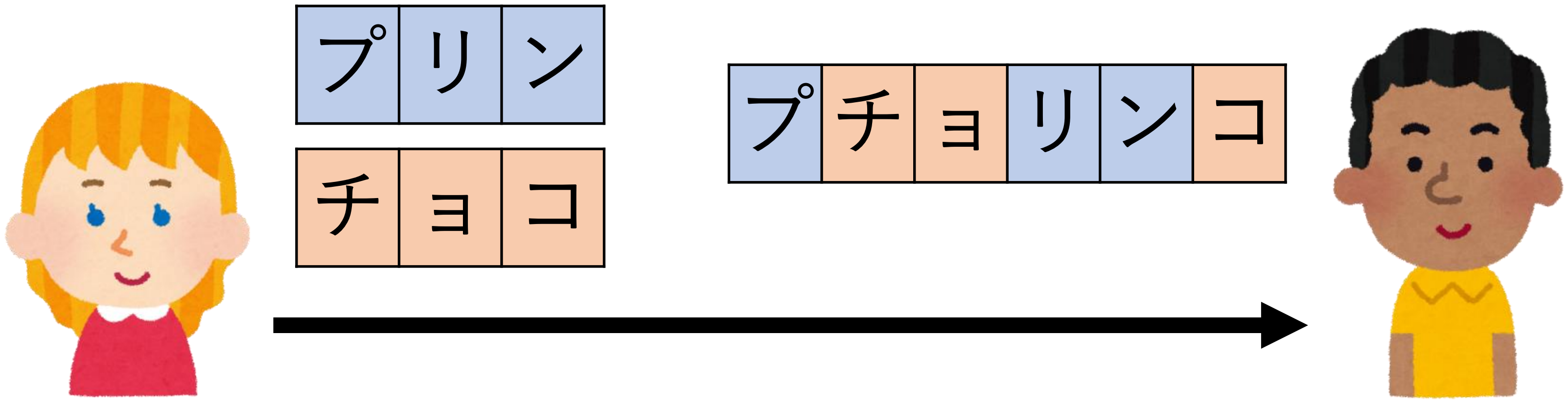
解説: 平木 康傑

問題概要

- Anna が Bruno に整数を伝言したいが，伝達内容に制限がある
  - Anna は同じ長さのバイナリ列 2 つを送る
    - 長さは 1 以上 2000 以下．上限を L とおく
  - Bruno はそれを「混ぜた」バイナリ列を受け取り，整数を復元する

「混ぜ方」の概要

- ある |S| 文字をそのままの順番で抜きだすと S に一致し，
  残りの |T| 文字をそのままの順番で抜きだすと T に一致する



ケツメイシ『譲れない事』歌い出し

- 時代がどんなに大きく変わっても
  変わらないものはあるんだって信じて
  時代が自由を奪おうとしても
  譲れない事があるんだ

不変量に注目しよう

- どんな混ぜ方をしても一定になる量を見つければ,
  それを利用して伝えたいものを表せる
- 例えばどんなものがある？

不変量の例

- 1 の個数

不変量の例

- 1 の個数
  - たとえば Anna が (1111, 1100) を送れば，それをどう混ぜても Bruno が受け取る数列には必ず 1 がちょうど 6 個入っている
  - 0 個から 2L = 4000 個までの 2L + 1 = 4001 通りを表せて，小課題 1 は解ける (5 点)

不変量の例

- 1 の個数
- 長さ
    - こっちでも小課題 1 は解ける

不変量の例

- 1 の個数
- 長さ
  - この 2 つは組み合わせられる
  - (長さ, 1 の個数) の組は全部で 3 + 5 + 7 + … + (2L+1) = L^2 + 2L 通り
  - 上限 L = 2000 なので 4,000,000 までをカバーできた
  - これで小課題 2 も通る (+5 → 10 点)

不変量の例

- 1 の個数
- 長さ
- 端の文字
  - S, T の先頭がともに 1 の場合，混ぜても先頭は 1
  - S, T の先頭がともに 0 の場合，混ぜても先頭は 0
  - S, T の末尾がともに 1 の場合，混ぜても末尾は 1
  - S, T の末尾がともに 0 の場合，混ぜても末尾は 0

不変量の例

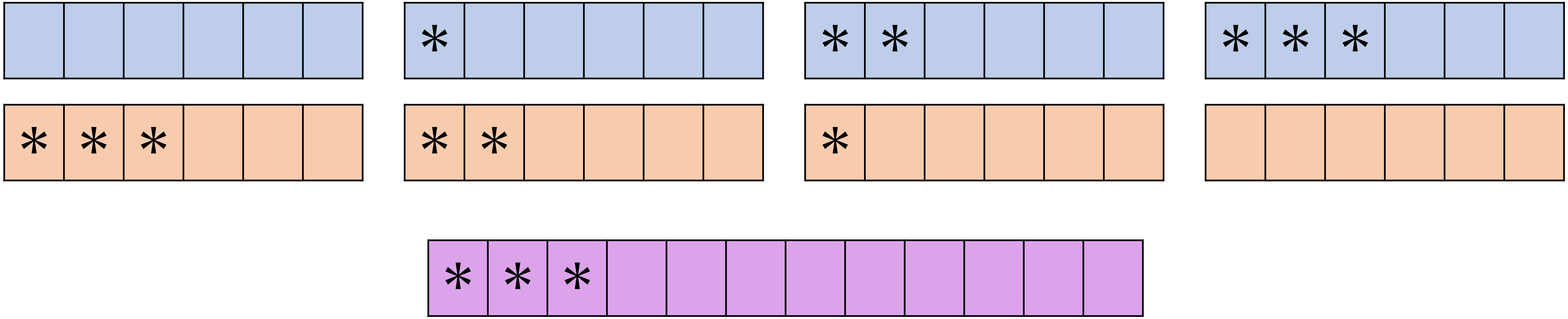
- 1 の個数
- 長さ
- 端の文字
  - S, T の {先頭, 末尾} がともに [0, 1] の場合，混ぜても{〃}は [〃]
  - 少々工夫はいるが，これを使って表せる範囲を 4 倍にできる
  - これで小課題 3 も通る (+3 → 13 点)

不変量の例

- 1 の個数
- 長さ
- 端の文字 ← 1 文字しか使わないなんてもったいない！
  - ~~S, T の {先頭, 末尾} がともに [0, 1] の場合，混ぜても{〃}は [〃]~~
  - → ______の場合，どう混ぜても先頭の____文字のうち 1 は____個

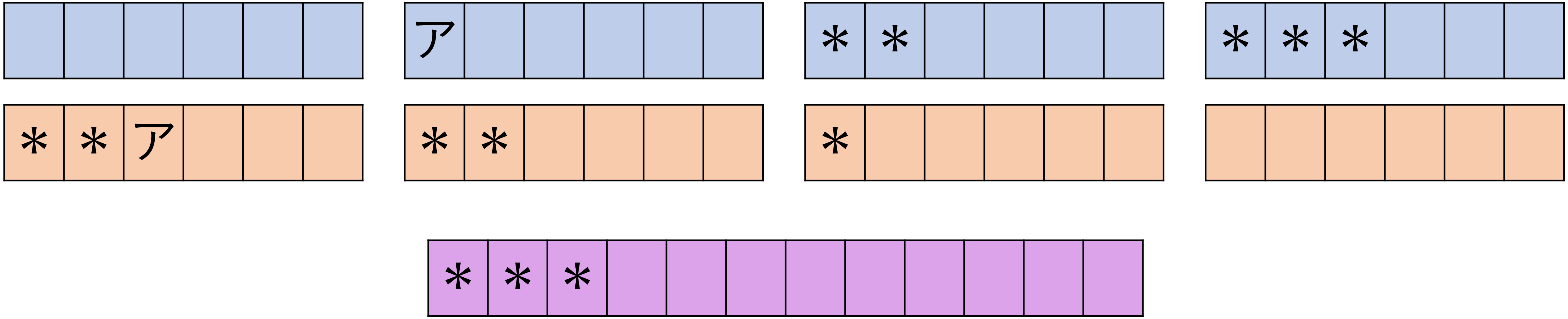
累積和が不変量になるような構成

- 端の____文字
  - ______の場合，どう混ぜても先頭の____文字のうち 1 は____個
- 例えば，先頭の 3 文字に選ばれる組合せは以下の通り



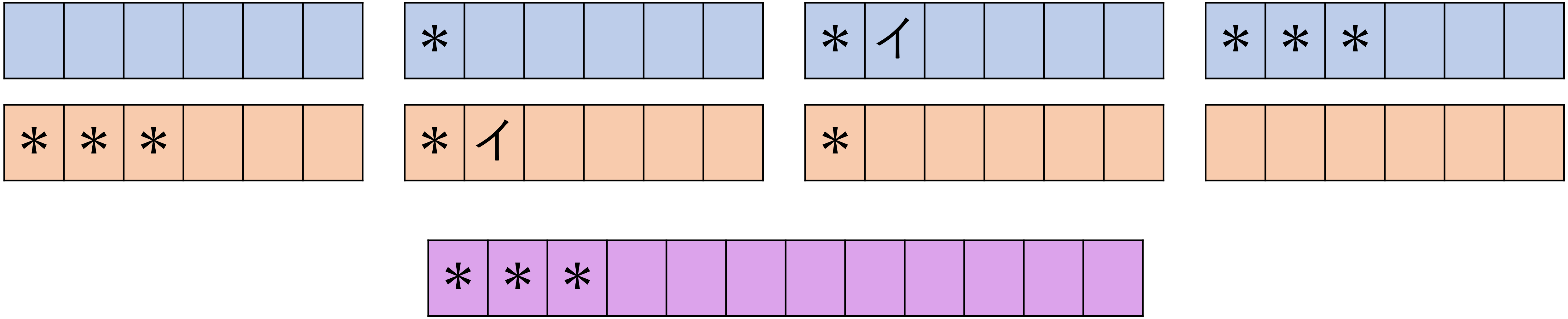
累積和が不変量になるような構成

- 端の____文字
  - ______の場合，どう混ぜても先頭の____文字のうち１は____個
- 例えば，先頭の３文字に選ばれる組合せは以下の通り



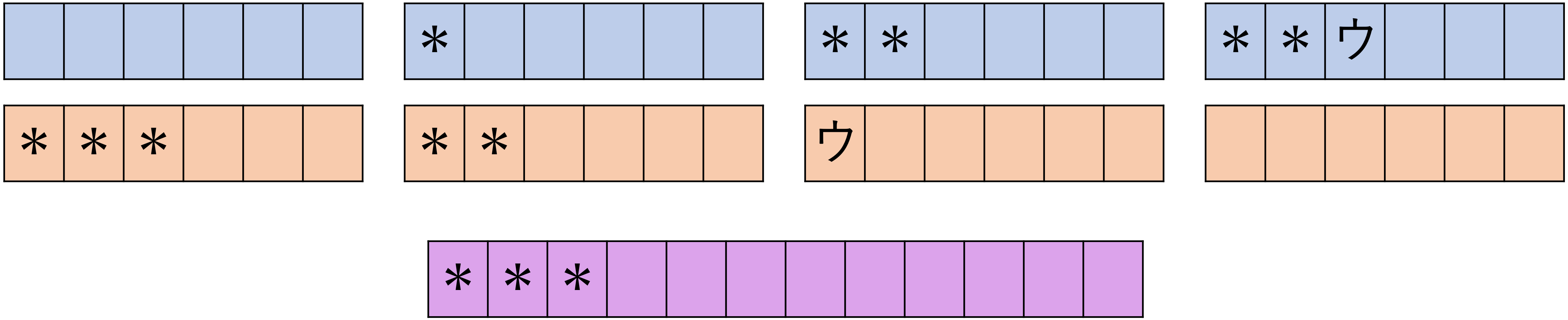
累積和が不変量になるような構成

- 端の____文字
  - ______の場合，どう混ぜても先頭の____文字のうち１は____個
- 例えば，先頭の３文字に選ばれる組合せは以下の通り



累積和が不変量になるような構成

- 端の____文字
  - ______の場合，どう混ぜても先頭の____文字のうち 1 は____個
- 例えば，先頭の 3 文字に選ばれる組合せは以下の通り



累積和が不変量になるような構成

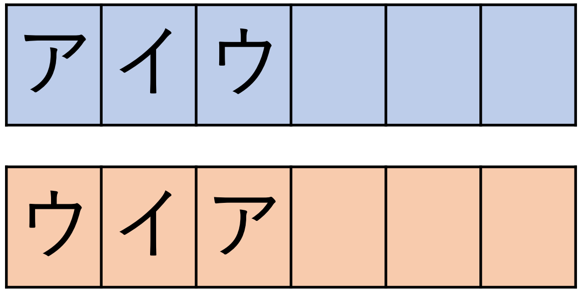
- 端の_____文字
  - _______の場合，どうまぜても先頭の_____文字のうち１は_____個
- 先頭３文字に選ばれる組合せの差分を取ることで，
  以下のア，イ，ウがそれぞれ同じなら
  先頭３文字に含まれる１の個数も一定とわかる

ア	イ	ウ			

ウ	イ	ア			

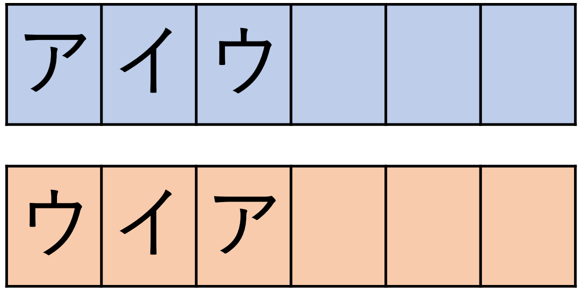
累積和が不変量になるような構成

- 端の____文字
  - ______の場合，どう混ぜても先頭の____文字のうち 1 は____個
- 以下のア，イ，ウがそれぞれ同じなら
  先頭 3 文字に含まれる 1 の個数も一定とわかる
- 一般に，S の先頭 k 文字 = reverse(T の先頭 k 文字) のとき，
  どう混ぜても
  先頭 k 文字に含まれる 1 の個数は一定



累積和が不変量になるような構成

- S の先頭 k 文字 = reverse(T の先頭 k 文字) の場合,
  どう混ぜても先頭の k 文字に含まれる 1 の個数は一定
- たとえば S = “11100…”, T = “00111…” とすれば,
  どう混ぜても先頭 5 文字に 1 はちょうど 3 つ含まれる



- 末尾についても同様のことが言える

不変量の重ねがけ

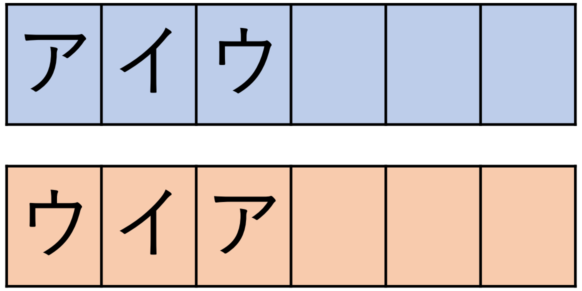
- 「長さ $|S|$」「先頭 $|S|/2$ 文字」「末尾 $|S|/2$ 文字」を組み合わせる
  - 長さ $|S|$ に対して,
    「先頭」で $|S| / 2$ 通り,
    「末尾」で $|S| / 2$ 通り,
    掛けあわせて $|S|^2 / 4$ 通りの不変量を伝えられる
  - $|S| = 1, 2, \ldots, 2000$ で合わせて
    $2000^3 / 12$ $(+\alpha)$ 通りになる
    - つまり 6.6×10^8 通り
  - 小課題 4 も通る (+12 → 25 点)

ア	イ	ウ			

ウ	イ	ア			

不変量の重ねがけ

- 「長さ |S|」「先頭 |S|/3 文字」「末尾 |S|/3 文字」さらに「全体の 1 の個数」を組み合わせる
  - 長さ |S| に対して,
    「先頭」で |S| / 3 通り,
    「末尾」で |S| / 3 通り,
    「全体の 1 の個数」で 2 |S| / 3 通り,
    あわせて 2/27 |S|^3 通りの不変量を伝えられる
  - |S| = 1, 2, ..., 2000 で合わせて 1/54 |S|^4 (+α) 通り
  - 小課題 5 も通る (+15 → 40 点)



不変量の重ねがけ

- 「先頭 k 文字」そのものをたくさん重ねがけすることもできる
- DP + 経路復元によって長さごとの最適な配置を求めることで, L = 937 で 10^18 まで区別できる
- +23 → 63 点

ア	イ	ウ	エ	オ	ア	イ	ウ

ウ	イ	ア	オ	エ	ウ	イ	ア

落ち着こう

- 「先頭 k 文字」を 1 回重ねがけるごとに長さが倍増するので, 140 には届きそうにない

落ち着こう

- 「先頭 k 文字」を 1 回重ねがけるごとに長さが倍増するので，140 には届きそうにない
- 少し見方を変えて，「+1/-1 としたときの累積和」に注目してみる

+1/-1 累積和

- +1 と -1 を交互に置いた数列で先頭から累積和をとるとどうなる？

+1/-1 累積和

- +1 と -1 を交互に置いた数列で先頭から累積和をとるとどうなる？
- -1, 0, 1 のどれかになる

+1/-1 累積和

- +1 と -1 を交互に置いた数列で先頭から累積和をとるとどうなる？
- -1, 0, 1 のどれかになる
- つまり，０からのズレが高々 1

高々 1 だけズレる

- T を [+1, -1, +1, -1, ...] として，S に割り込んだときの累積和は？

高々 1 だけズレる累積和

- T を [+1, -1, +1, -1, …] として，S に割り込んだときの累積和は？
- 純粋に S の要素だけだったときと比べて，高々 1 しかズレない

高々 1 だけズレる累積和

- T を [+1, -1, +1, -1, ...] として，S に割り込んだときの累積和は？
- 純粋に S の要素だけだったときと比べて，高々 1 しかズレない
- つまり，S が十分声を張れば T はかき消されると考えられる！

ノイズの入る累積和

- $S = 1 + \sum_{k=1}^{60} b_k \times$ (直前と同じビットは 2 個，異なるビットは 3 個)
- $\mathrm{T} = 10101010 \ldots$
- 最初のビットを読んで「直前のビットは 1 だった」ことにする
- 以下を 60 回繰り返す
  - +1/-1 累積和を始めて，+2 か -2 になったら停止してビットを特定する

ノイズの入る累積和

L=181 なので，累計 80 点

- $S = 1 + \sum_{k=1}^{60} b_k \times$ (直前と同じビットは 2 個，異なるビットは 3 個)
- $\mathrm{T} = 10101010 \ldots$
- 最初のビットを読んで「直前のビットは 1 だった」ことにする
- 以下を 60 回繰り返す
  - +1/-1 累積和を始めて，+2 か -2 になったら停止してビットを特定する
- 停止時点で，直前のビット b_{k-1} に対し，以下のいずれかが成り立つ
  - S のビットがちょうど読まれ，T の次のビットは b_{k-1} と同じ
  - S のビットが 1 つだけ残り，T の次のビットは b_{k-1} と異なる

+「長さ」+「長さに収まる最適な割り当て」

- $S = 1 + \sum_{k=1}^{???} b_k \times$ (直前と同じビットは 2 個，異なるビットは 3 個)
- $T = 10101010 \ldots$

L=142 なので，累計 96 点

- 最初のビットを読んで「直前のビットは 1 だった」ことにする
- 以下を ~~60 回繰り返す~~ 長さが尽きるまで繰り返す
  - +1/-1 累積和を始めて，+2 か -2 になったら停止してビットを特定する
- 停止時点で，直前のビット b_{k-1} に対し，以下のいずれかが成り立つ
  - S のビットがちょうど読まれ，T の次のビットは b_{k-1} と同じ
  - S のビットが 1 つだけ残り，T の次のビットは b_{k-1} と異なる

＋「長さ」＋「長さに収まる最適な割り当て」

＋「端の文字」

- $S = 1 + \sum_{k=1}^{???} b_k \times$ (直前と同じビットは 2 個，異なるビットは 3 個)

L=142 なので，100 点

- T = 10101010 ...

～2

- 最初のビットを読んで「直前のビットは 1 だった」ことにする
- 以下を ~~60 回繰り返す~~ 長さが尽きるまで繰り返す
  - +1/-1 累積和を始めて，+2 か -2 になったら停止してビットを特定する
- 停止時点で，… 以下のいずれかが成り立つ
  - … ちょうど読まれ，T の次のビットは b_{k-1} と同じ
  - S のビットが 1 つだけ残り，T の次のビットは b_{k-1} と異なる

伝える整数が奇数なら，送受信するビットを全て反転する

+「長さ」+「長さ…る最適な割り当て」

得点分布

+「端の文字」

0 点: 1 人

1 − $\sum_{k=1}^{??} b_k$ × (直前と同じビットは 2 … 異な… ビ…

5 点: … 1 人

1 3 点: 7 人

- T = 10…

点: 12 人

5 点: 4 人

…40 なので,

- 最初のビットを読んで「直前のビット = 1 だった」ことにする
- 以下を ~~60 回繰り返す~~ 長さが尽きるまで繰り返す
  - +1 … 積和を求めて，+2 か -2 になったら停止 … ビットを特定する

40 点: 1 人

- 停止時点で … かが成り立つ
  - … ちょうど読まれ, … のビットは b_{k-1} と同じ
  - S のビットが 1 つだけ残り，T の次のビットは b_{k-1} と異なる

伝える整数が奇数なら，送信するビットを全て反転する

80 点: 3 人

100 点: 0 人