JOI 2024/25 春 Day 1

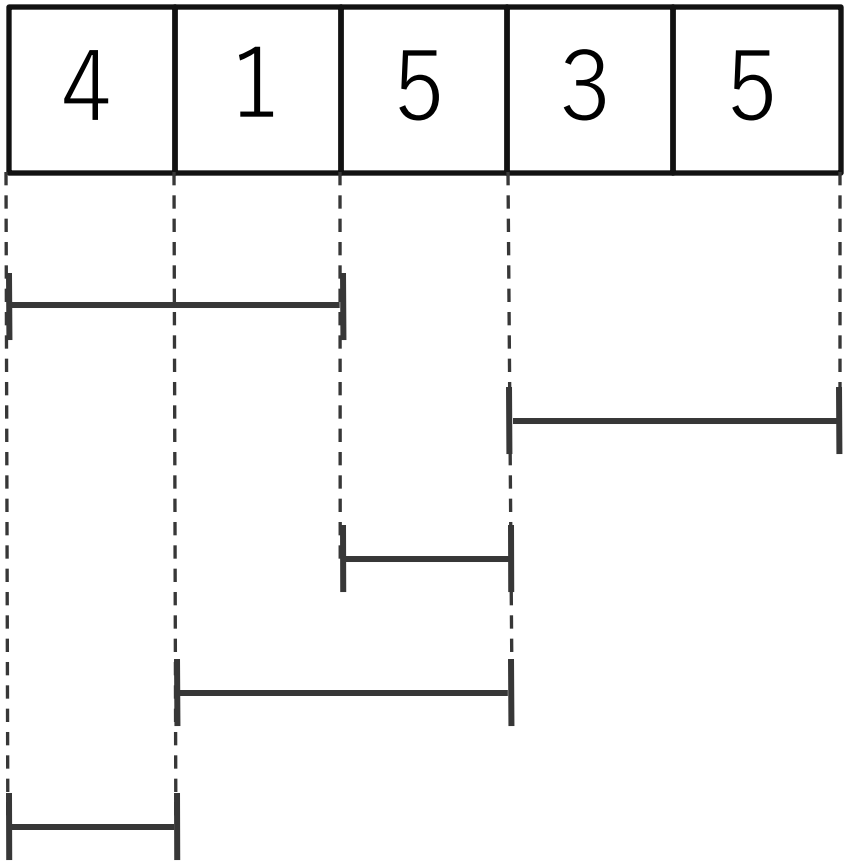
展覧会 3
(Exhibition 3)

解説：渡邉雄斗 (yuto1115)

問題概要

問題概要

- 長さ N の数列 $A = (A_1, A_2, \ldots, A_N)$ が与えられる
- $B_j\ (1 \leq j \leq M) \coloneqq \max\{A_{L_j}, A_{L_j+1}, \ldots, A_{R_j}\}$ と定義する
- A をうまく並び替えることで得られる、辞書順最大の数列 $B = (B_1, B_2, \ldots, B_M)$ を出力せよ



B_j

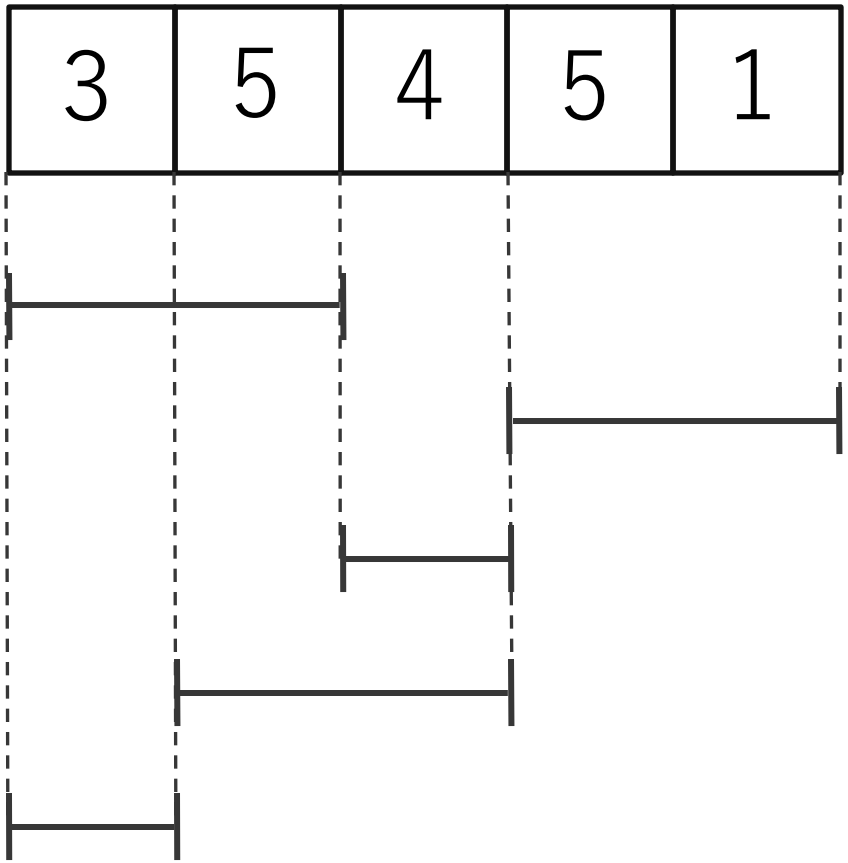
4

5

5

5

4



B_j

5

5

4

5

3

Step 1: 多項式時間で解く

小課題 1 $N \leq 400,\ M \leq 400$ (19 pts.)

Step 1: 多項式時間で解く

- 辞書順最大化なので、前から貪欲
- こんな問題を解きたくなる：
  - $B_1, \ldots, B_{j-1}$ の値が既に確定している（可能だと分かっている）。

    B_j の値を $k\ (1 \leq k \leq N)$ とすることは可能だろうか？

Step 1: 多項式時間で解く

- B_j の値を k $(1 \leq k \leq N)$ とすることは可能だろうか？
  - 値が既に k と確定した区間たちに区間 $j = [L_j, R_j]$ を追加した集合を考える
  - 各区間内に k が少なくとも一個は含まれなければならない
  - $c_k =$ (A に含まれる k の個数) として、c_k 個以下の添字 $(1 \sim N)$ をうまく選ぶことで、各区間内に選んだ添字が一つ以上含まれるようにできる必要がある
  - これは **区間スケジューリング問題の答えが c_k 以下** であることと同値
- これは十分条件（証明略）

Step 1: 多項式時間で解く

- いくつかの記法を定義する
  - $S_i\ (1 \leq i \leq N)$ ： $B_j = i$ となることが確定した区間 j の集合
  - $IS(S)$：区間の集合 S に対し、S から選べる、互いに交わりを持たない区間の個数の最大値（区間スケジューリング問題）

Step 1: 多項式時間で解く

- アルゴリズム：
  - $S_1, S_2, \ldots, S_N$ を空集合で初期化する
  - $j = 1,2, \ldots, M$ の順に以下を行う
    - $IS(S_i \cup \{j\}) \leq c_i$ を満たす **最大** の i を求め、B_j をその値にセット
    - S_{B_j} に区間 j を追加する
- 計算量：$\mathrm{O}(\mathrm{MN} + M \sum_{i=1..N} |S_i| \log |S_i|) = \mathrm{O}(\mathrm{MN} + \mathrm{M}^2 \log \mathrm{M})$

Step 2: IS 高速化

小課題 2 $N \leq 400,\ M \leq 10^5$ (9 pts.)

小課題 3 $N, M \leq 10^5,\ A_i \leq 5$ (19 pts.)

Step 2: IS 高速化

- 小課題 2 ($N \leq 400, M \leq 10^5$) を解くには高速化が必要
  - 特に、$M^2 \log M$ の項をどうにかしたい
  - 区間スケジューリング問題を、区間の数ではなく N に依存する計算量で解きたい

Step 2: IS 高速化

- 区間スケジューリング問題を異なる視点から捉える：
  - $N+1$ 頂点 $(0, 1, \ldots, N)$ からなる有向グラフがある
  - 各 i $(0 \leq i \leq N-1)$ に対し、$i \to i+1$, 長さ 0 の辺がある
  - 各区間 $[l, r)$ (0-indexed) に対し、$l \to r$, 長さ 1 の辺がある
  - 頂点 0 から頂点 N への **最長経路** の長さは？

Step 2: IS 高速化

- $Len(i,\ s,\ t) \coloneqq$ 区間集合 S_i に対応するグラフにおいて、頂点 $s \to t$ への最長経路の長さ
- S_i に 区間 $[l,\ r)$ を新たに追加可能 $\Leftrightarrow Len(i,\ 0,\ l) + 1 + Len(i,\ r,\ N) \leq c_i$
- $Len(i,\ 0,\ k)$ および $Len(i,\ k,\ N)$ をすべての $i,\ k\ (1 \leq i \leq N, 0 \leq k \leq N)$ について管理
    - ある区間 j に対し、
        - B_j の値を求めるのは $O(N)$
        - $Len\big(B_j,\ 0,\ k\big), Len(B_j,\ k,\ N)$ の更新は $O(N)$
    - よって、全体で $O(N^2 + MN)$

Step 2: IS 高速化

- $K = \max A_i$ として、K が小さい場合の解法を考える
  - 現状だと $O(NK + MK + MN)$（左から順に初期化、S_j の求値、Len の更新）
- 小課題 3 $(N, M \leq 10^5, K \leq 5)$ を解くには高速化が必要
  - $Len(i,\ 0,\ k), Len(i,\ k,\ N)$ の更新に時間がかかりすぎる
  - 一方で、S_j の求値には計算量的に余裕がある
    - ⇒ **平方分割**

Step 2: IS 高速化

- その前に、余分な区間を消しておきたい
  - 他の区間を完全に内包しているような区間はムダなので、消す
    - 残った区間の集合 $[l_0, r_0), [l_1, r_1), \ldots, (l_i < l_{i+1},\ r_i < r_{i+1})$ は set で管理できる
  - すると、最長経路は貪欲で求まるようになる
    - 現在地を p として、p を左端にもつ区間 (に対応する辺) があるなら
      それを使う、ないなら $p \to p + 1$ と進む

Step 2: IS 高速化

- D: バケットサイズ
  - $[0, N]$ を $[0, D), [D, 2D), [2D, 3D), \ldots$ と N/D 個のバケットに分割
- $To(i,\ s)$ ≔ 区間集合 S_i に対応するグラフにおいて、頂点 s からスタートして、**s と異なるバケットに到達するまで** 貪欲に進み続けたときの、到達点
- $Len'(i,\ s)$ ≔ 頂点 s から $To(i,\ s)$ まで貪欲に進んだときの経路の長さ

Step 2: IS 高速化

- $Len(i,\ s,\ N)$ を求めるには？
  - $s \to To(i,\ s) \to To\big(i,\ To(i,\ s)\big) \to \cdots$ と進み続ければ、N/D 回以内に頂点 N に到達
  - この過程で通る辺の長さの総和は $Len'(i,\ s) + Len'\big(i,\ To(i,s)\big) + \cdots$ で求まる

Step 2: IS 高速化

- $\mathrm{B_j}$ の値が求まった。S_{B_j} に区間 j を追加する際に必要な更新操作は？
  - set をごにょごにょして、「余分な区間をすべて取り除いた後に残る区間集合」の差分を求める（グラフの問題のおける辺の追加・削除に相当）
  - ある辺 $l \to r$ の追加・削除は、**l と異なるバケットに属する頂点 s に対する $To(i,s), Len'(i,s)$ の値には影響を及ぼさない**
    - l と同じバケット内の To, Len' は $O(D)$ かけて一から求め直す
  - 一度削除されたら追加されないので、更新回数は合計 $O(M)$

Step 2: IS 高速化

- 初期化：$O(NK)$
- 各 j に対して、
  - S_j の求値：$O(K \cdot \frac{N}{D})$
  - To, Len の更新：ならし $\mathrm{O}(\log N + D)$
- よって、$D = \sqrt{N}$ のとき全体で $O\left(NK + MK\sqrt{N}\right)$

  $D = \sqrt{NK}$ のとき全体で $O\left(NK + M\sqrt{NK}\right)$

Step 3: 大きい N への対応

小課題 4 $N, M \leq 10^5$, $A_i = i$ (12 pts.)

小課題 5 $N, M \leq 10^5$, $c_i \leq 5$ (17 pts.)

Step 3: 大きい N への対応

- $N \leq 10^5$ のとき、各 j に対して「S_i に区間 j を追加できるか」を $i = N, N-1, \ldots, 1$ と順にチェックしていくことはできない
- まずは $A_i = i$ (subtask 4) という特殊ケースについて考える
  - 特に、すべての i について $c_i = 1$

Step 3: 大きい N への対応

- $l_i \coloneqq \mathrm{S_i}$ に含まれる区間の**左端**の**最大**値

  $r_i \coloneqq \mathrm{S_i}$ に含まれる区間の**右端**の**最小**値
- このとき、$IS(S_i) \leq 1 \Leftrightarrow l_i < r_i$
- したがって、S_i に区間 $[l', r')$ を追加可能

  $$\Leftrightarrow \max\{l_i, l'\} < \min\{r_i, r'\}$$

  $\Leftrightarrow [l_i, r_i)$ と $[l', r')$ が共通部分を持つ

Step 3: 大きい N への対応

- S_i に区間 $[l', r')$ を追加可能 ⇔ $[l_i, r_i)$ と $[l', r')$ が共通部分を持つ
- よって、$B_j = [l_i, r_i)$ と区間 j が共通部分を持つ最大の i
- 以下を満たす配列 $x = (x_0, x_1, \ldots, x_{N-1})$ を考える：
  - 各 i について、$x_{l_i} = x_{l_i+1} = \cdots = x_{r_i} = i$
  - それ以外の x の要素の値はすべて -1
- このとき、B_j は区間 j 内に含まれる x の要素の最大値

  ただし最大値が -1 のときは、S_i が空である最大の i

Step 3: 大きい N への対応

- B_j が求まったあと、$[l_{B_j}, r_{B_j})$ がどのように変化するかは簡単に分かり、

  それに伴う配列 x の更新は区間代入操作に該当する
- よって、区間代入・区間 max が高速に処理できれば良い
- 遅延セグメント木を用いれば、全体で $O(N + M \log N)$
- なお、異なる i に対する $[l_i, r_i)$ が共通部分を持つことはない（証明略）

Step 3: 大きい N への対応

- $c_i = 1$ とは限らない、一般のケースについて考えたい

Step 3: 大きい N への対応

- 各 S_i について、以下を定める：
  - $k_i \coloneqq IS(S_i)$
  - S_i に対する区間スケジューリング問題を解く際の貪欲法について考えたとき、
    - 左に詰めるように貪欲に取った区間の列（を左から右に並べたもの）
      $$\left[\alpha_1^{(i)}, \beta_1^{(i)}\right), \left[\alpha_2^{(i)}, \beta_2^{(i)}\right), \ldots, \left[\alpha_{k_i}^{(i)}, \beta_{k_i}^{(i)}\right)$$
    - 右に詰めるように貪欲に取った区間の列（を左から右に並べたもの）
      $$\left[\gamma_1^{(i)}, \delta_1^{(i)}\right), \left[\gamma_2^{(i)}, \delta_2^{(i)}\right), \ldots, \left[\gamma_{k_i}^{(i)}, \delta_{k_i}^{(i)}\right)$$
    - $\left[l_t^{(i)}, r_t^{(i)}\right)\ (1 \leq t \leq k_i) \coloneqq \left[\gamma_t^{(i)}, \beta_t^{(i)}\right)$

Step 3: 大きい N への対応

- 各 S_i について、$\left[l_t^{(i)}, r_t^{(i)}\right)\ (1 \leq t \leq k_i) \coloneqq \left[\gamma_1^{(i)}, \beta_1^{(i)}\right)$
- ここで、以下が成立する (証明略) :
  - S_i に区間 $[l,' r')$ を新たに追加しても $IS(S_i)$ の値が変化しない

    ⇔ ある t $(1 \leq t \leq k_i)$ に対して、$\left[l_t^{(i)}, r_t^{(i)}\right)$ と $[l', r')$ が共通部分を持つ
- $A_i = i$ のケースの一般化になっている

  ⇒ 同様に遅延セグメント木を用いたアルゴリズムを設計できる！

Step 3: 大きい N への対応

- 説明のため、以下のように用語を定義：
  - ある $i\ (1 \leq i \leq N)$ に対し、$IS(S_i) = c_i$ であるとき S_i は **飽和している** といい、そうでないとき **未飽和である** という
  - 未飽和な S_i のうち、i が最大であるものを **フロンティア** と呼ぶ
- フロンティアを S_f とすると、以下が成立する (証明略)：
  - すべての $i > f$ について、S_i は飽和している
  - すべての $i < f$ について、S_i は空である

Step 3: 大きい N への対応

- アルゴリズム：

1. 飽和した S_i たちに対する $\left[l_t^{(i)}, r_t^{(i)}\right)$ を管理するための遅延セグメント木を用意する
2. $j = 1,2,\ldots,M$ の順に以下を行う：

   I. 遅延セグメント木に対する区間 max クエリを用いて、飽和している S_i のうち区間 j を追加しても $IS(S_i)$ が変わらないものがあるか判定し、あるならばそのうち i が最大なものを求める

   II. 飽和集合 S_i に追加できる場合：$\left[l_t^{(i)}, r_t^{(i)}\right)$ を計算し直し、適切に遅延セグメント木への区間代入操作を行う

   III. できない場合：フロンティア S_f に区間 j を追加する。S_f が飽和したか確かめ、したならば S_{f-1} を新たにフロンティアとする

Step 3: 大きい N への対応

- 計算量を考える。ボトルネックは以下の通り：
  - 飽和集合 S_i に追加する場合の $\left[l_t^{(i)}, r_t^{(i)}\right)$ の再計算
  - フロンティア S_f に追加する場合の $IS(S_f)$ の再計算
- 余分な区間 (他の区間を包含している区間) を逐次取り除いた上で、残った区間を set 等で適切に管理すれば、どちらも $O(\max c_i \cdot \log M)$ で可能
- $F = \max c_i$ として、全体で $O(N + M(F \log M + \log N))$

Step 4: 統合

小課題 6 $N, M \leq 10^5$ (18 pts.) (満点)

Step 4: 統合

- 小課題 5 解法のボトルネックは以下の通りであった：
  - 飽和集合 S_i に追加する場合の $\left[l_t^{(i)}, r_t^{(i)}\right)$ の再計算 … $O(F \log M)$
  - フロンティア S_f に追加する場合の $IS\left(S_f\right)$ の再計算 … $O(F \log M)$

Step 4: 統合

- 小課題 5 解法のボトルネックは以下の通りであった：
  - 飽和集合 S_i に追加する場合の $\left[l_t^{(i)}, r_t^{(i)}\right)$ の再計算 … $O(F \log M)$
  - フロンティア S_f に追加する場合の $IS(S_f)$ の再計算 … $O(F \log M)$
- 本当にすべてを再計算し直す必要はあるのか？ない
  - 左に詰めて取った区間の列 $\left[\alpha_1^{(i)}, \beta_1^{(i)}\right), \left[\alpha_2^{(i)}, \beta_2^{(i)}\right), \ldots, \left[\alpha_{k_i}^{(i)}, \beta_{k_i}^{(i)}\right)$ の更新について考える
  - 新たに $[l, r)$ を追加するとする。 $\beta_t^{(i)} \leq l < \beta_{t+1}^{(i)}$ なる t をとる

Step 4: 統合

- 左に詰めて取った区間の列 $\left[\alpha_1^{(i)}, \beta_1^{(i)}\right), \left[\alpha_2^{(i)}, \beta_2^{(i)}\right), \ldots, \left[\alpha_{k_i}^{(i)}, \beta_{k_i}^{(i)}\right)$
  の更新について考える
- 新たに $[l, r)$ を追加するとする。 $\beta_t^{(i)} \leq l < \beta_{t+1}^{(i)}$ なる t をとる
- $\alpha_u^{(i)}, \beta_u^{(i)}$ $(u \leq t)$ の値は変化しないのでスルー
- 新たな $\alpha_u^{(i)}, \beta_u^{(i)}$ の値を $u = t+1, t+2, \ldots,$ の順に求める。
  ある u に対して値が変化しなかったら、その後の処理は打ち切って良い
- 計算量は改善しているのか？

Step 4: 統合

- 改善していて、ならし $O(\log M)$ になっている
- 「左/右に詰めて取った区間の列」から一度削除された区間はもう二度とこの列に現れないことから従う
    - 列から削除された区間が再び列に現れるとき、$IS(S_i)$ の値が 1 以上増加している。これは S_i が飽和していることに矛盾

Step 4: 統合

- 小課題 5 解法のボトルネックは以下の通りであった：
  - 飽和集合 S_i に追加する場合の $\left[l_t^{(i)}, r_t^{(i)}\right)$ の再計算 … ならし $O(\log M)$
  - フロンティア S_f に追加する場合の $IS(S_f)$ の再計算 … $O(F \log M)$

Step 4: 統合

- 小課題 5 解法のボトルネックは以下の通りであった：
  - 飽和集合 S_i に追加する場合の $\left[l_t^{(i)}, r_t^{(i)}\right)$ の再計算 … ならし $O(\log M)$
  - フロンティア S_f に追加する場合の $IS(S_f)$ の再計算 … $O(F \log M)$
- これも同様に「左/右に詰めて取った区間の列」を管理し、変化がなくなった時点で更新を打ち切ればならし $O(\log M)$ になるのでは？
  - ならない
  - 未飽和のとき、各区間は $O(c_i)$ 回この列から出し入れされうる

Step 4: 統合

- しかし、未飽和の区間集合のうち区間が追加されうるのはフロンティア S_f ただ一つ
- 区間集合一つだけなら平方分割で管理できる
  - ただし、フロンティアが移るたびに $O(N)$ の初期化はできないことに注意
  - 変更が加わった箇所をすべて保持しておいて、その部分だけ元に戻せば良い
- 全体計算量：$O(N + M(\sqrt{N} + \log M))$

おわりに

- 実装が大変
  - サンプルコードは 300 行ほど
  - 区間を扱う問題は頭が壊れがちですが、頑張りましょう
- 証明もちょっと大変
- 平方分割パートは Link-Cut Tree で代用可：全体 $O(N + M\,(\log M + \log N))$
  - 代用？

得点分布

得点	小課題	人数
40	1, 2, 4	2
31	1, 4	1
28	1, 2	2
19	1	8
12	4	1
0	ø	12