JAG ICPC模擬地区予選2020
J: X-percent blooming

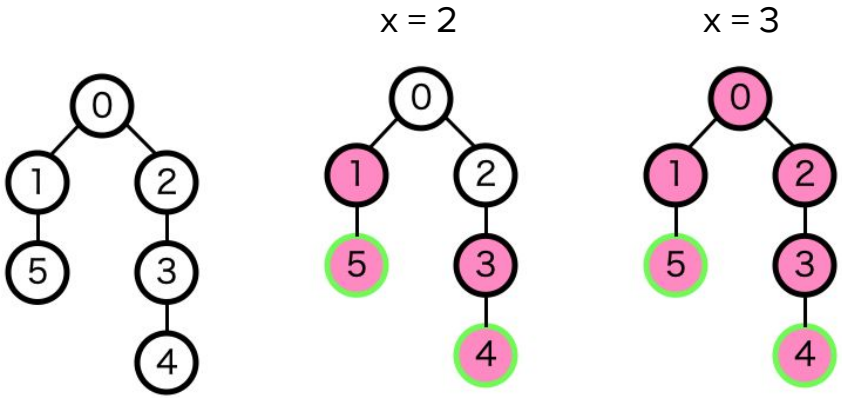
原案: Darsein
解法改善: yosupo
問題文: Darsein
データセット: not
解答: Darsein
解説: Darsein

問題概要

初期状態で根のノード1つのみからなる木がある。以下のクエリを処理せよ

1. 指定されたノードの子として新たなノードを1つ追加
2. 距離d以内の子孫のうち1つ以上が葉であるノードの個数をb(d)とする。
   事前に与えられた固定値X, Yについて、b(X) / b(Y) を求める

制約: $1 \leq Q \leq 10^5$



考察

問題の言い換えを考える

- クエリを先読みして全てのノードを持つ木を構築する
- 各ノードを白く塗り、根だけを黒く塗っておく (黒 ⇔ 葉)
- ノードを追加する操作は以下のように言い換えられる
  - 構築済みの木の中で対応したノードを黒く塗る
  - その親が黒ければ白く塗る
- L(x) := 自分から距離 x 以内の子孫のうち、黒いノードの個数 とおく
- 観測は以下のように言い換えられる
  - L(x) > 0のノードの個数を数える

解法

言い換えを元に、木に対して以下の操作ができればよい

・あるノードからx個先の祖先までのノードすべての値を+1 (= 黒く塗る)

・同様にx個先の祖先まで -1 (= 白く塗る)

・L(v) = 0となるvを数える (#{L(v) > 0の頂点} = N - ${L(v) = 0の頂点})

➔ これらは Heavy-Light Decomposition + 遅延更新Segment Tree などで処理可能

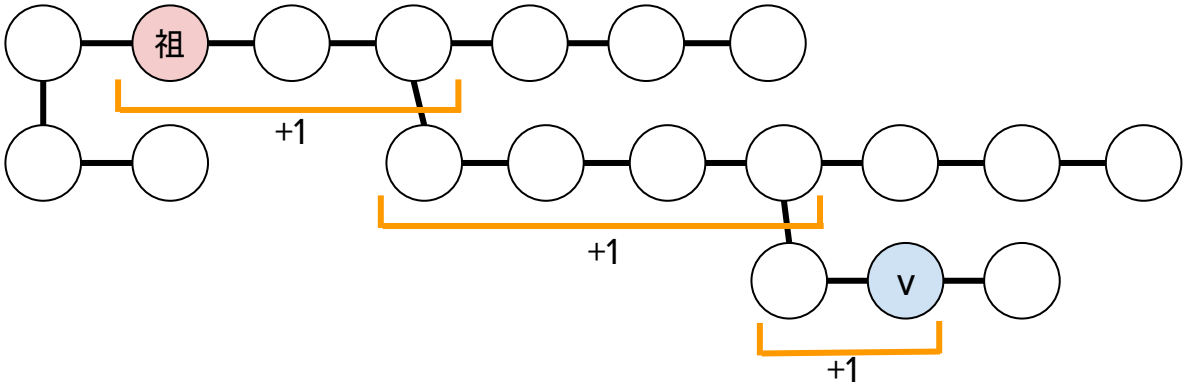
解法: Heavy-Light Decomposition

Heavy-Light Decompositionの各列をSegment Treeで管理

・値を足すとき: 祖先へと遡りながら区間に+1 / -1

・0を数えるとき: グローバルな変数で全体の個数を管理、観測クエリではその値を返す
各列の値を更新するときに更新前後の差分を全体の個数に足す

➔ 範囲addと0の個数を数えられるSegment TreeがあればOK



距離 x = 8

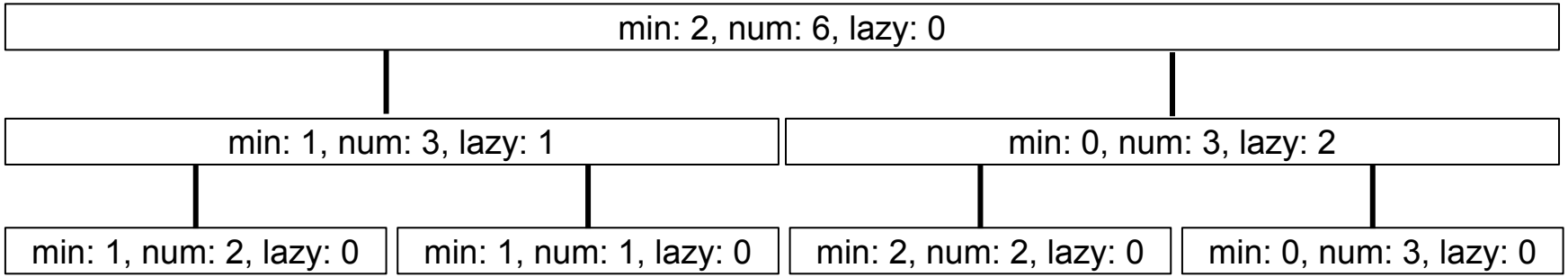
解法: Segment Tree

0の個数を数える ← 非負値の中の最小値とその個数を数えられればOK

- 最小値が非ゼロ → 0は0個
- 最小値がゼロ → 0の個数は最小値の個数

範囲addと最小値の個数を管理するには遅延更新Segment Treeを使えばOK

- 範囲が一致するとき加算値を lazy に一時保存、最小値は min + lazy

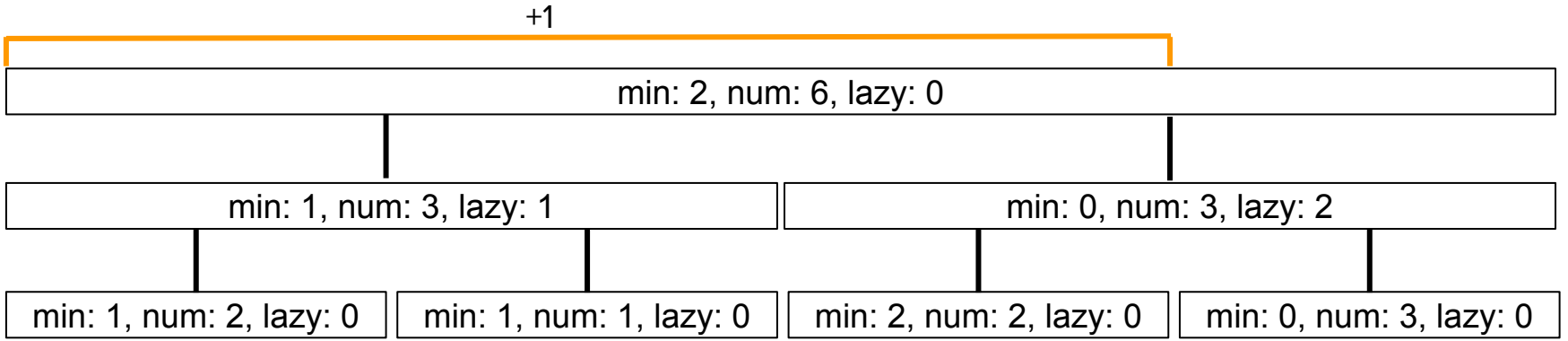


解法: Segment Tree

0の個数を数える ← 非負値の中の最小値とその個数を数えられればOK

- 最小値が非ゼロ → 0は0個
- 最小値がゼロ → 0の個数は最小値の個数

範囲addと最小値の個数を管理するには遅延更新Segment Treeを使えばOK

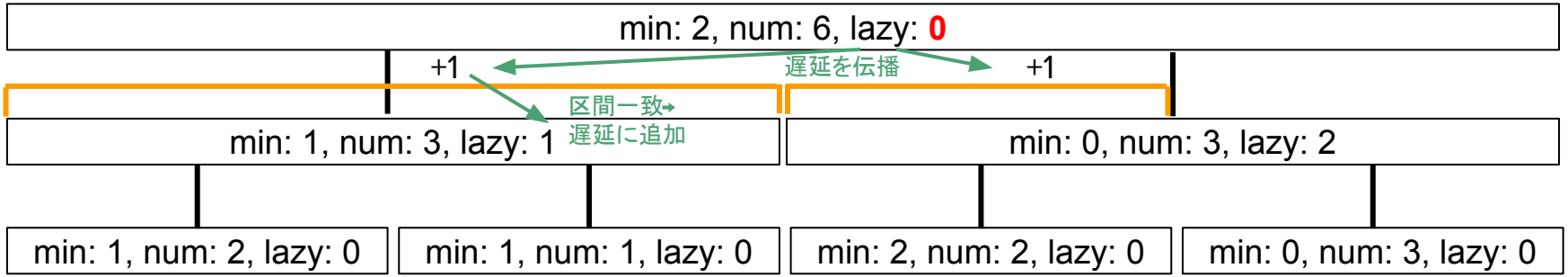


解法: Segment Tree

0の個数を数える ← 非負値の中の最小値とその個数を数えられればOK

- 最小値が非ゼロ → 0は0個
- 最小値がゼロ → 0の個数は最小値の個数

範囲addと最小値の個数を管理するには遅延更新Segment Treeを使えばOK

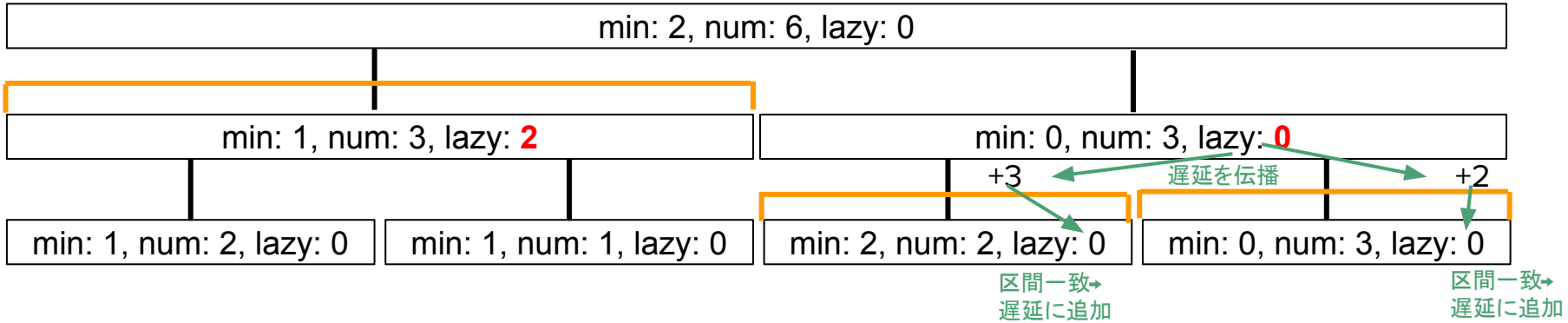


解法: Segment Tree

0の個数を数える ← 非負値の中の最小値とその個数を数えられればOK

- 最小値が非ゼロ → 0は0個
- 最小値がゼロ → 0の個数は最小値の個数

範囲addと最小値の個数を管理するには遅延更新Segment Treeを使えばOK

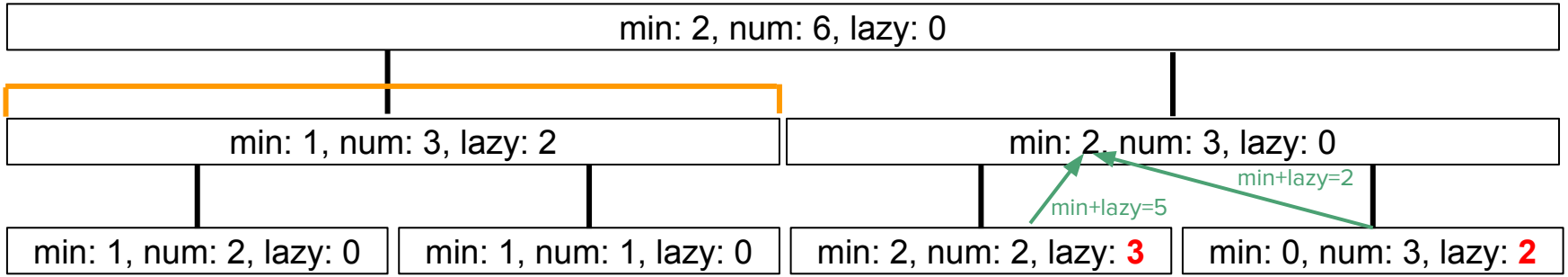


解法: Segment Tree

0の個数を数える ← 非負値の中の最小値とその個数を数えられればOK

- 最小値が非ゼロ → 0は0個
- 最小値がゼロ → 0の個数は最小値の個数

範囲addと最小値の個数を管理するには遅延更新Segment Treeを使えばOK

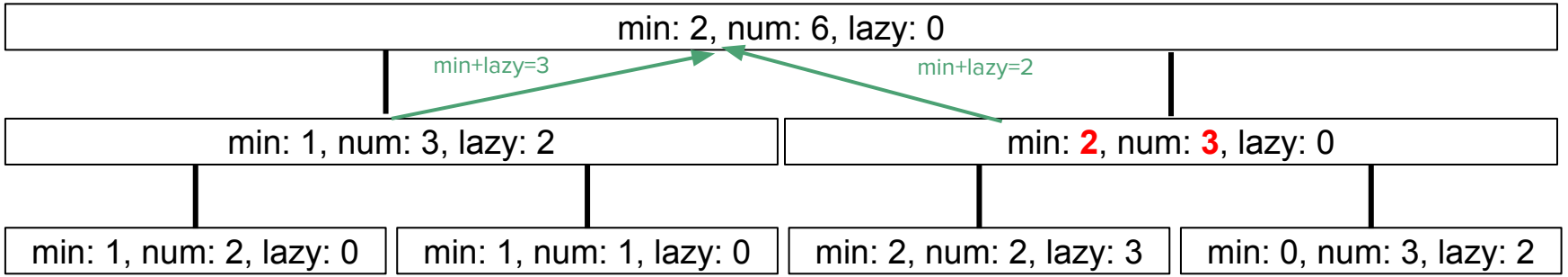


解法: Segment Tree

0の個数を数える ← 非負値の中の最小値とその個数を数えられればOK

- 最小値が非ゼロ → 0は0個
- 最小値がゼロ → 0の個数は最小値の個数

範囲addと最小値の個数を管理するには遅延更新Segment Treeを使えばOK

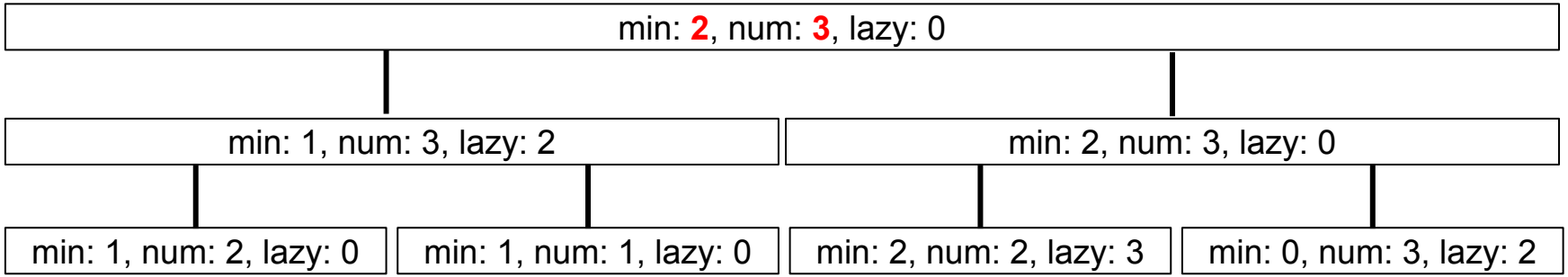


解法: Segment Tree

0の個数を数える ← 非負値の中の最小値とその個数を数えられればOK

- 最小値が非ゼロ → 0は0個
- 最小値がゼロ → 0の個数は最小値の個数

範囲addと最小値の個数を管理するには遅延更新Segment Treeを使えばOK



解法: 計算量

- 最終的な木のノード数 N: O(Q)
- Heavy-Light Decompositionの構築: O(N)
- 各列のSegment Treeの構築: O(N)
- Q回のクエリについて、
  - 距離xの祖先まで値を加算 : $O(\log^2 N)$
    - Heavy-Light Decompositionの高さはO(logN)なので、高々O(logN)個の列に対してO(logN)のSegment Tree上のクエリ
  - 0の個数をカウント : O(1)
- 全体で $O(N + Q\log^2 N) = O(Q\log^2 Q)$

統計

Acceptances / Submissions

- 5 / 10 (50.00%)

First Acceptance

- __KING__ (01:56)