JOI ツアー 2 (joitour)

2026/03/22 春 Day 2

解説：保坂 結 (hos.lyric)

問題概要

- N 頂点の木
  - 頂点に値段 $A_u \in [1, N]$ が書いてある
- M 本のパス
- Q 通りの予算 $B_q \in [1, 2N]$
- 各予算について，以下のような選び方を数えよ
  - パスを選ぶ
  - →パス上の異なる 2 頂点 u, v を選ぶ
  - →値段の和がちょうど予算になっている $(A_u + A_v = B_q)$
- $N \leq 100\,000$，$M \leq 200\,000$，$Q \leq 2\,000$

小課題 1 ($N, M, Q \leq 100$)

- 全探索
  - M 本のパスについて，通る頂点を列挙
  - 頂点の 2 つ組を列挙
  - 予算を添え字とする配列に記録
  - $O(N^2 M + Q)$ 時間

小課題 2, 3 ($N \leq 5\,000$)

- 頂点の 2 つ組を全列挙できる
- 2 頂点 u, v を両方通るパスを高速に数えられればよい
  - パスグラフのとき
    - $u < v$ として，左端 $\in [1, u]$，右端 $\in [v, N]$ という条件
    - 2 次元累積和
  - 一般の木のとき
    - Euler tour で長方形クエリ ≤ 2 回にできる
- $O(N^2 + M + Q)$ 時間・$O(N^2 + M + Q)$ 空間

小課題 4, 5 ($Q = 1$)

- 予算 B が決まっている
- ひとまず，パス 1 本あたり $O(N)$ で解くには？

パス上の各頂点 `u` に対し:

```
answer += count[B - A[u]]
count[A[u]] += 1
```

- 1 頂点追加したとき答えの差分を $O(1)$ 時間で計算できる
- 1 頂点削除もできる (逆に操作するだけ)

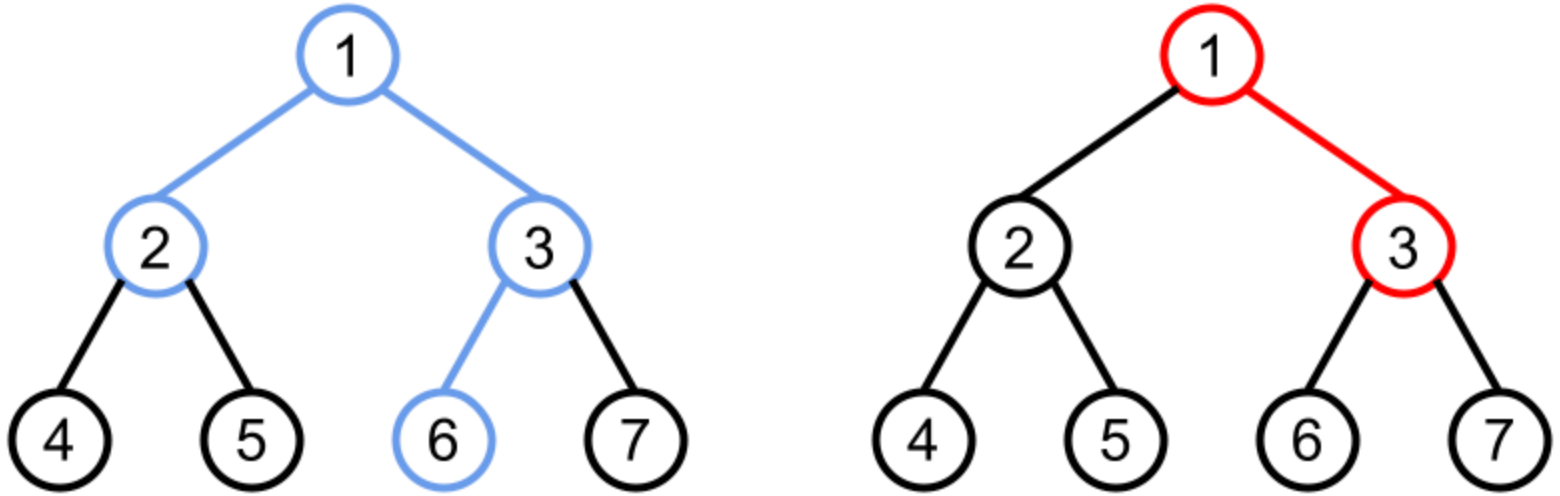
小課題 4 ($Q = 1$，パスグラフ)

- 列なら **Mo のアルゴリズム**が適用可能
  - 列をブロックサイズ L ごとに分ける
  - M 個の区間を (左端の属するブロック, 右端) でソート
  - 区間の伸縮回数が左 $O(ML)$・右 $O(N^2/L)$
  - $L = \Theta(N/\sqrt{M})$ ととると伸縮回数が $O(N\sqrt{M})$
- 「現在の区間に対する答え」を持って更新していくことで，各区間についての答えが求まり，それを合計して出力
- $O(N\sqrt{M} + M)$ 時間・$O(N + M)$ 空間

小課題 5 ($Q = 1$)

- 木のパスにも Mo が応用可能
- **Euler tour** (頂点列) $u_0, \ldots, u_{2N-2}$ をとる (u_i と u_{i+1} が隣接)
- パス s-t を，$(u_l, u_r) = (s, t)$ または $(u_l, u_r) = (t, s)$ である区間 $[l, r]$ に対応させる
  - １通りとは限らないがどれでもよい
- 区間の伸縮 $\longleftrightarrow$ パスの端点を隣に動かす $\longleftrightarrow$ パスの伸縮
  - 区間を伸ばしてもパスが縮むこと (やその逆) はある
- 対応させた区間を Mo の順にソートすれば，パスの伸縮回数が $O(N\sqrt{M})$ になる

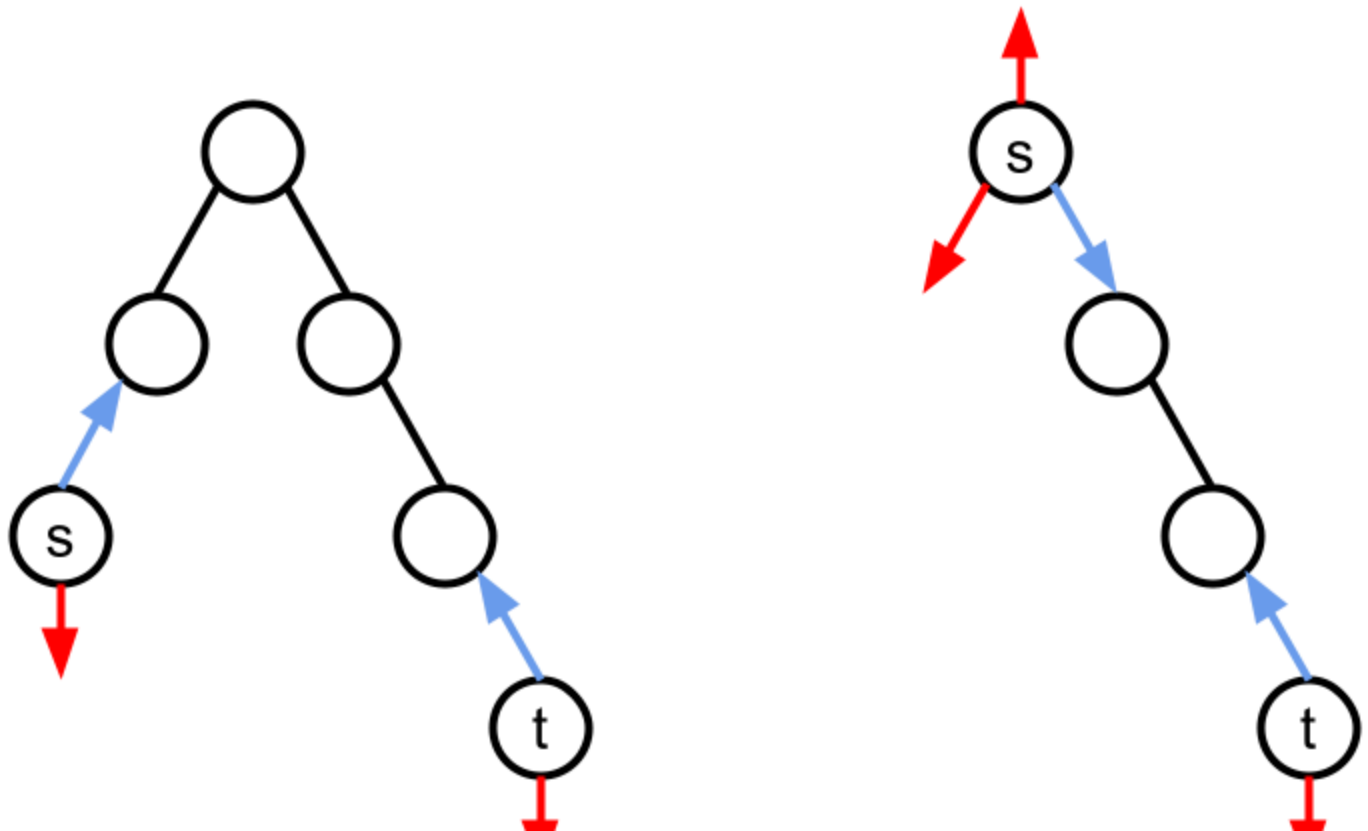
小課題 5 ($Q = 1$)



1 2 4 2 5 2 1 3 6 3 7 3 1

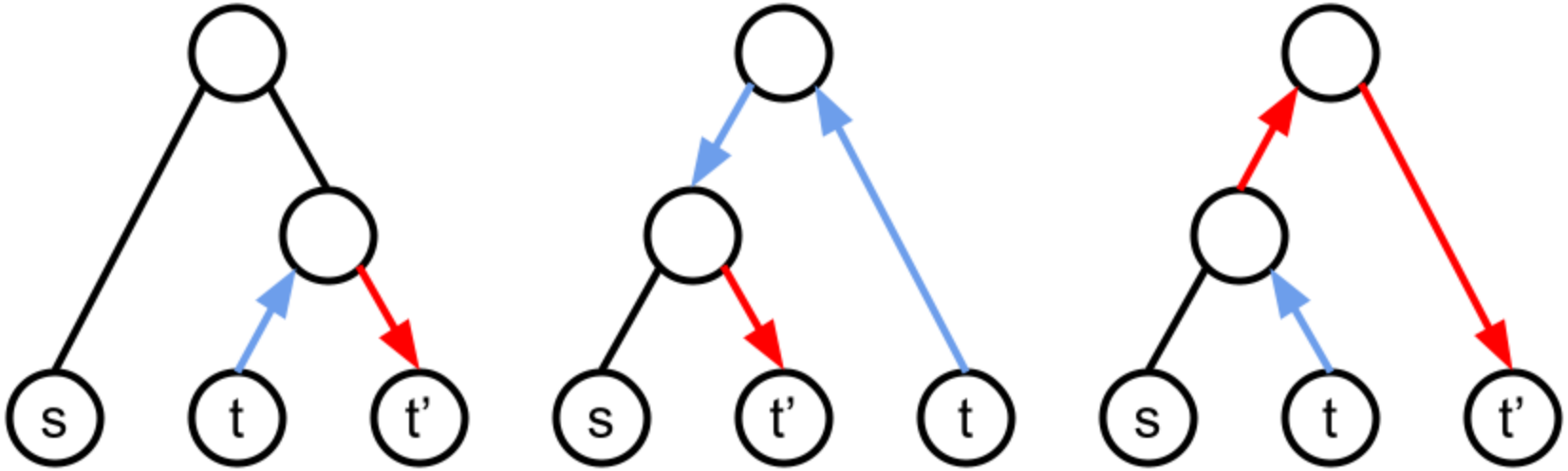
小課題 5 ($Q = 1$)

- パスの伸縮を管理する方法 (その 1)
  - 先祖・子孫関係にあるかを $O(1)$ 時間で判定できるとよい
  - Euler tour で最初と最後の登場を記録するとできる



小課題 5 ($Q = 1$)

- パスの伸縮を管理する方法 (その 2)
  - Euler tour 上の区間を 1 ずつ伸縮させなくてもよい
  - パスを s-t から s-t' にするとき，t から t' への最短路に沿って動かす
  - $\mathrm{LCA}(s,t), \mathrm{LCA}(s,t'), \mathrm{LCA}(t,t')$ の一致関係で場合分け



考察整理タイム

別バージョンの問題

- 入力
  - N 頂点の木，頂点重み A_u
  - M 本のパス
  - 重み $W[1], \ldots, W[2N]$，そのうち非 0 なのは Q か所 $W[B_q]$
- 出力
  - パスを選んでパス上の異なる 2 頂点 u, v を選ぶ方法について，$W[A_u + A_v]$ の合計 (値 1 個だけ)

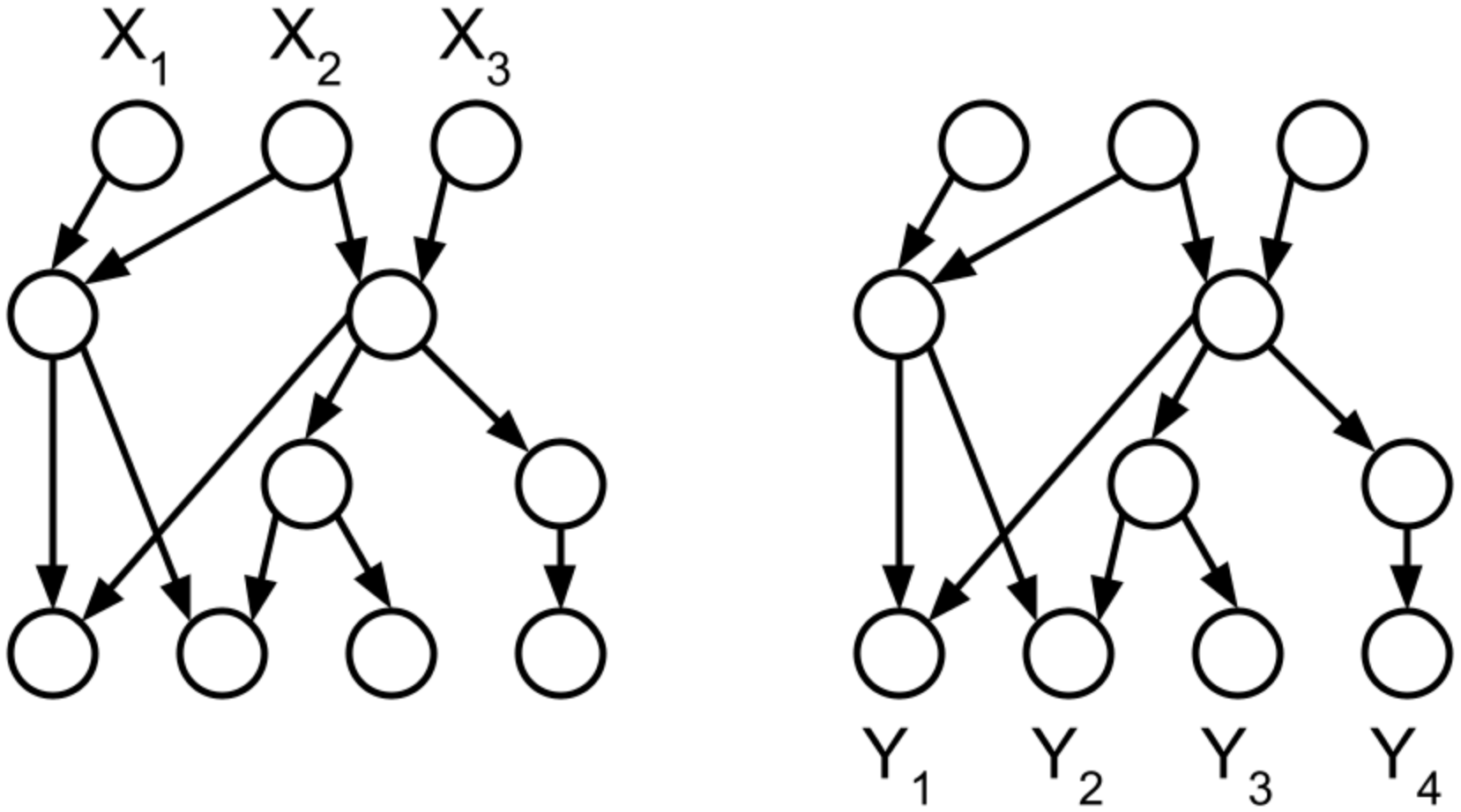
考察整理タイム

- 何がうれしいの？
- **転置原理**によって，別バージョンの問題が解ければ，元の問題の (計算量がほぼ同じ) 解法が機械的に得られることが示せる
- 「本質的な難しさ」みたいなものは変わらないですが，本問では，人によっては別バージョンの問題のほうが少し考えやすいと思います
- この問題の満点解法を理解するのに必須ではありませんが，考察テクニックのひとつとして紹介します

転置原理

- 辺重み付き DAG が与えられる
- 頂点 $S_1, \ldots, S_P$ は入次数 0 (始点)
- 頂点 $T_1, \ldots, T_Q$ は出次数 0 (終点)
- S_p から T_q へのパスの重みの合計を $C_{p,q}$ とする
  - パスの重み：通る辺の重みの積
- 問 1：$X_1, \ldots, X_P$ が与えられる．各 q に対し $\displaystyle\sum_{p=1}^{P} X_p C_{p,q}$ を求めよ
- 問 2：$Y_1, \ldots, Y_Q$ が与えられる．各 p に対し $\displaystyle\sum_{q=1}^{Q} C_{p,q} Y_q$ を求めよ

転置原理



転置原理

- 転置原理：問 1 と問 2 の関係にある問題はほぼ同じ計算量で解ける
  - 片方の問題だけ空間計算量が節約できてしまうことはある
- 問 1 の解法：始点側から DP
- 問 2 の解法：終点側から DP
- 解法の変換
  - 同じ変数を用意する
  - 入力以外を 0 で初期化する
  - 手順を逆向きに見ていく
  - 変数 `f`, `g` と係数 `c` に対して，`g += c * f` を `f += c * g` に変換

転置原理

- 典型的な適用例：$\mathrm{ans}_1, \ldots, \mathrm{ans}_Q$ を求めよ，という形の問題
- 重み X が与えられて，$X \cdot \mathrm{ans}_1, \ldots, X \cdot \mathrm{ans}_Q$ を求めよ，という問題とみなす
- 転置すると：重み $Y_1, \ldots, Y_Q$ が与えられて，$\sum_{q=1}^{Q} \mathrm{ans}_q \cdot Y_q$ を求めよ
- 実際にアルゴリズムを変換するには
  - `g += c * f` の形のステップに分解して逆に辿って変換する (大変)
  - 何段階かに区切って，↑のイメージを持つことで触る変数や計算量を意識しながら，意味を考えて解く (おすすめ)

転置原理

- ある意味絶対に「必要」にならないテクニック
- 問題を変換して，簡単に思えることもそうでないこともあります
- しかし解けない問題に帰着してしまうことがないので，割と考え得
- 多項式関連の問題や線型代数を学ぶと深く理解できると思いますが，発想自体は情報オリンピックの範囲でも有用です
- 参考資料
  - Alin Bostan, Grégoire Lecerf, and Éric Schost. "Tellegen's principle into practice."
  - 転置原理まとめ | Mathenachia
  - "転置原理" で web 検索

考察整理タイム

転置した問題 (再掲)

- 入力
  - N 頂点の木，頂点重み A_u
  - M 本のパス
  - 重み $W[1], \ldots, W[2N]$，そのうち非 0 なのは Q か所 $W[B_q]$
- 出力
  - パスを選んでパス上の異なる 2 頂点 u, v を選ぶ方法について，$W[A_u + A_v]$ の合計 (値 1 個だけ)

考察整理タイム

転置した問題

- 頂点集合 U, V に対し，値 $U * V$ を次で定める
  - $u \in U$ と $v \in V$ の組に対する $W[A_u + A_v]$ の合計
- 性質
  - $(U \pm U') * V = (U * V) \pm (U' * V)$
  - $U * (V \pm V') = (U * V) \pm (U * V')$
    - 集合の $\pm$ が適切に定義できるとき

小課題 6 ($M \leq 1\,000$，パスグラフ) 転置

- 区間 $[l, r)$ に対する答えを以下の和に分解 (小課題 4, 5 でやった発想)
  - $[l, l) * \{l\}$
  - $[l, l+1) * \{l+1\}$
  - $\vdots$
  - $[l, r-1) * \{r-1\}$
- 以降頂点の添え字を 0 から始めます
- $[l, r) * \{v\} = [0, r) * \{v\} - [0, l) * \{v\}$ を用いて，$[0, u) * \{v\}$ という形のもの $O(NM)$ 個になる

小課題 6 ($M \leq 1\,000$，パスグラフ) 転置

- 分解した $[0, u) * \{v\}$ たちを u の昇順に見ていくことにする
- 目標
  - 各 u に対して，$O(Q)$ 時間くらいかけて何か準備していい
  - 各 u に対していろいろな v が来たとき，$O(1)$ 時間くらいで答えたい
- 「$A_v = a$ のときの $[0, u) * \{v\}$」を各 $a \in [1, N]$ について持っておけたらうれしい
- それは $[0, u)$ のときと $[0, u + 1)$ のときでの変化が少ない
  - $W[A_u + a]$ だけ増えるので，Q か所だけ更新
- $O(NM + NQ)$ 時間・$O(NM)$ 空間 (MLE)

小課題 6 ($M \leq 1\,000$，パスグラフ) 転置

- $[0, u) * \{v\}$ に分解した結果をすべて明示的に持つと MLE
  - $[l, l) * \{l\} = [0, l) * \{l\} - [0, l) * \{l\}$
  - $[l, l+1) * \{l+1\} = [0, l+1) * \{l+1\} - [0, l) * \{l+1\}$
  - $\vdots$
  - $[l, r-1) * \{r-1\} = [0, r-1) * \{r-1\} - [0, l) * \{r-1\}$
- まとめて持とう
  - $[0, u) * \{u\}$ 型は N 種類しかないので個数をカウント
  - 残りは $[0, l) * [l, r)$ という形でまとめて記録
- $O(N + M + Q)$ 空間

考察整理タイム

元の問題

- 頂点集合 U, V に対し，長さ Q の列 $U * V$ を次で定める
  - q 項目は，$u \in U$ と $v \in V$ の組であって $A_u + A_v = B_q$ なものの個数
- 性質
  - $(U \pm U') * V = (U * V) \pm (U' * V)$
  - $U * (V \pm V') = (U * V) \pm (U * V')$
    - 集合の $\pm$ が適切に定義できるとき
    - 右辺の $\pm$ は項ごと

小課題 6 ($M \leq 1\,000$，パスグラフ)

- (機械的に転置をとれる人は転置をとって終わりです)
- 転置した問題での方針
  - $[0, u) * \{v\}$ を u の昇順に見る
  - 各 u に対して，$O(Q)$ 時間かけて準備する
  - 各 u に対していろいろな v が来たとき，$O(1)$ 時間で答える
- 元の問題の方針
  - $[0, u) * \{v\}$ を u の降順に見る
  - 各 u に対して，$O(Q)$ 時間くらいかけて何か計算
  - 各 u に対していろいろな v が来たとき，$O(1)$ 時間くらいで何か準備

小課題 6 ($M \leq 1\,000$，パスグラフ)

- 答えを $O(NM)$ 個の $[0, u) * \{v\}$ たちに分解する
  - $[l, l) * \{l\}$, $[l, l+1) * \{l+1\}$, $\ldots$, $[l, r-1) * \{r-1\}$
- 頂点 u' が左側で関わる答えを $O(Q)$ 時間で計算するためにほしいものとは
  - 分解した $[0, u) * \{v\}$ たちのうち，$u' \in [0, u)$ (つまり $u' \leq u$) なものについての，A_v の分布 (長さ N の列)
- u' から $u'-1$ へ変わるときの分布の差は，いろいろな v が来て 1 箇所足すだけなので，1 個あたり $O(1)$ 時間
- $O(NM + NQ)$ 時間・$O(N + M + Q)$ 空間

小課題 10 (パスグラフ)

- ある程度少ない個数の $[0, u) * \{v\}$ に分解すれば解けるということ
- Mo の区間の伸縮とは，まさに $[l, r) * \{r\}$ や $[l, r) * \{l - 1\}$ の増減
- $k - 1$ 番目の区間から k 番目の区間への伸縮中に登場したものは，係数 $M + 1 - k$ で答えに寄与する $(1 \leq k \leq M)$
  - 係数付きの $[0, u) * \{v\}$ が $O(N\sqrt{M})$ 個になる
- メモリ節約も同様
  - $[0, u) * \{u\}$ 型や $[0, u] * \{u\}$ 型を分けておく
  - 残りを，区間 * 区間 の形にまとめる
    - この右側の区間の長さの和が $O(N\sqrt{M})$ で抑えられている
- $O(N\sqrt{M} + M + NQ)$ 時間・$O(N + M + Q)$ 空間

満点

- Mo のパスの伸縮によって，パス $*$ 頂点 という形に分解される
- 列の場合の $[0, u)$ に対応するものは，根付き木にして (頂点 0 を根とする)，根からのパス (根を 0 とする)
- パス s-t は，$\mathrm{LCA}(s, t) = l$ として，$[0, s] + [0, t] - [0, l] - [0, l)$ のように分解できる
  - 根からのパスを区間のように表記している
- 分解の情報を $O(N + M)$ 空間で持つのも同様
  - $[0, u] * \{u\}$ のようなものを分けておく
  - 残りを，根からのパス $*$ パス の形にまとめる

満点 転置

- 「$A_v = a$ のときの $[0, u] * \{v\}$」を各 $a \in [1, N]$ について持っておけたらうれしい
- 分解した $[0, u] * \{v\}$ たちを u について DFS しながら見ていく
  - u に潜るとき，Q か所増やす
  - u から上がるとき，Q か所減らす
- $O(N\sqrt{M} + M + NQ)$ 時間・$O(N + M + Q)$ 空間

満点

- u が関わる答えを $O(Q)$ 時間で計算するためにほしいものとは
  - 分解した $[0, u'] * \{v\}$ たちのうち，$u \in [0, u')$ (つまり u' が u の部分木内) なものについての，A_v の分布 (長さ N の列)
- $[0, u'] * \{v\}$ を u' について DFS 順に追加していくと，部分木は区間
  - u に潜るとき，答えから引く
  - u から上がるとき，答えに足す
- $O(N\sqrt{M} + M + NQ)$ 時間・$O(N + M + Q)$ 空間
  - LCA の計算で $O(N \log(N))$ や $O(M \log(N))$ 使っても OK

その他の小課題

- 小課題 7 ($M \leq 1\,000$)
  - Mo をせず，各パスを空集合から伸ばして作る
    - パスの伸縮での場合分けがなくなりちょっと楽
- 小課題 8, 9 ($N, M \leq 50\,000$)
  - メモリ節約をせず，区間 $*$ 点 への分解を明示的にすべて持つ
  - 満点解法だが定数倍が悪い場合
    - Mo のブロックサイズの決め方にも注意

得点分布

得点	人数	累積
24	1	1
17	1	2
13	2	4
12	1	5
9	1	6
7	6	12
4	1	13
3	6	19
0	11	30