KUPC2018 - M

整数と根付き木

原案：kazuma

問題概要

各ノードが配列を持っている根付き木に以下のクエリを行う

クエリ 1

- ある部分木の全ノードの配列に整数 x を k 個追加

クエリ 2

- ある頂点の配列内の整数で、y でXORして z 以下になるものの個数を取得

クエリ 3

- 根を変更する

クエリ３は後回し

根の変更？？？
- オイラーツアーを平衡二分木に入れる？
- Link-Cut Tree の evert 操作？

→どっちにしても大変

とりあえずクエリ３は置いといて、他のクエリだけで考える
- つまり根は１で固定とみなす

問題の言い換え

クエリ１が部分木に対する操作なので、オイラーツアーをとる

すると以下のような問題に言い換えられる

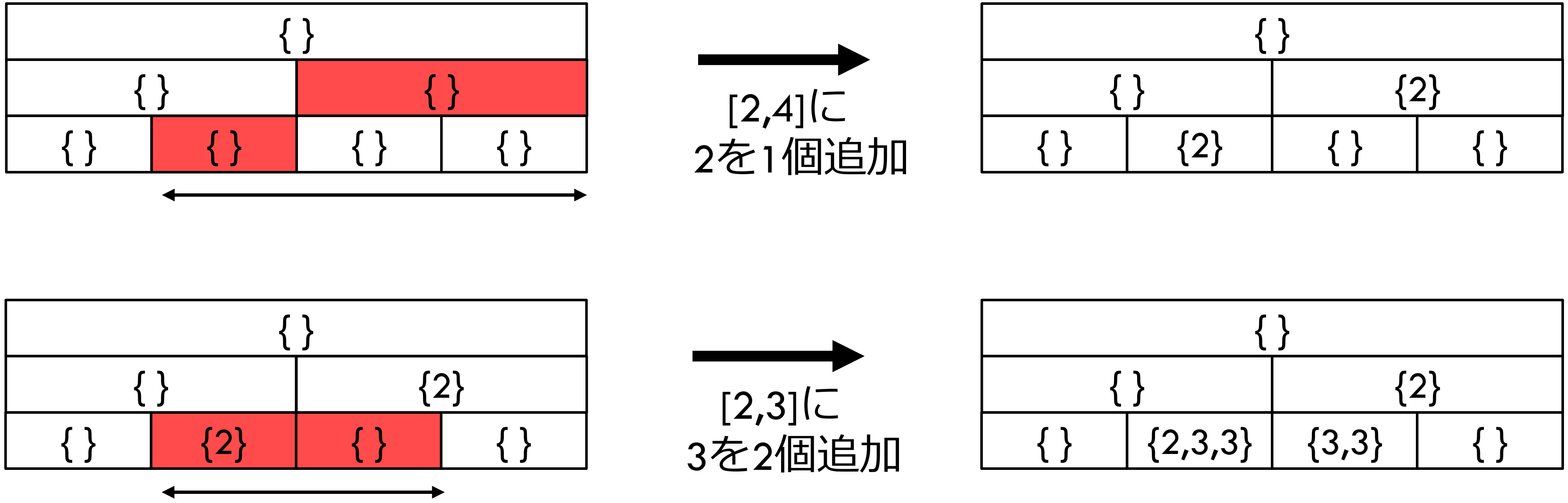
N個の配列が 1～N の順に並んでいて以下のクエリを行う

- クエリ１’
  - ある区間のすべての配列に x を k 個追加
- クエリ２’
  - ある位置の配列に y でXORして z 以下になる値がいくつあるかを求める

クエリ１’: 整数を追加する

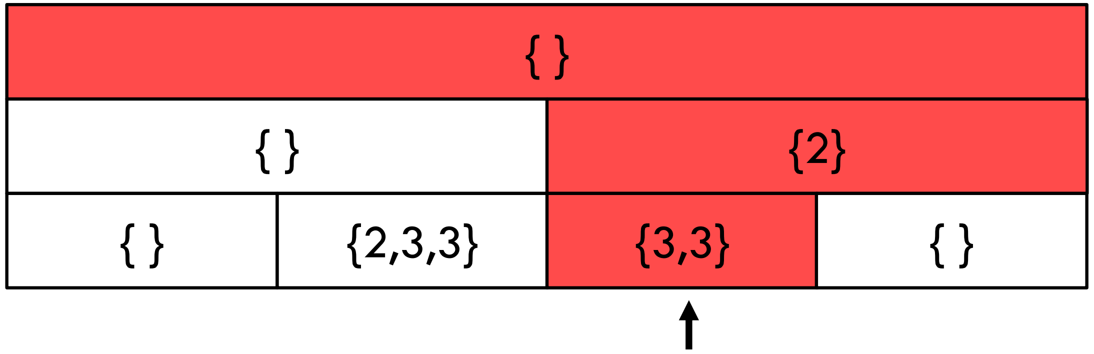
これはセグメントツリーのように管理すると可能

各ノードには値を追加できるような何らかのデータ構造(後述)をのせる



クエリ２’: 個数を求める

各配列は根からその位置までのノードをまとめたものになる



位置3の配列は {2,3,3}

XOR云々は後にして、とりあえず個数を求めたいなら logN 個のノードで求めてから足し合わせればよい

ノードにのせるもの

各ノードでは以下のクエリが要求される

- ある整数を k 個追加
- y でXORして z 以下になるものの個数

これは二分木の Trie を使うと $O(\log X)$ (Xは最大の要素) でできる

これでクエリ1, 2の両方とも $O(\log N \log X)$ になりOK

MLE

ただ、このままだと空間計算量が $O(Q \log N \log X)$ なので素直に実装したらギリギリMLEする(はず)

そこで、Trie を Patricia Trie にすると $O(Q \log N)$ になってAC

他に実装が楽な方法として、追加クエリを遅延評価するという手がある

- すると、各 Trie のノード数が追加クエリごとに高々 1 個しか増えないのでOK

クエリ３の対処法

ここまでクエリ３を無視してきたが、実は今の方針で根の変更も対処できる

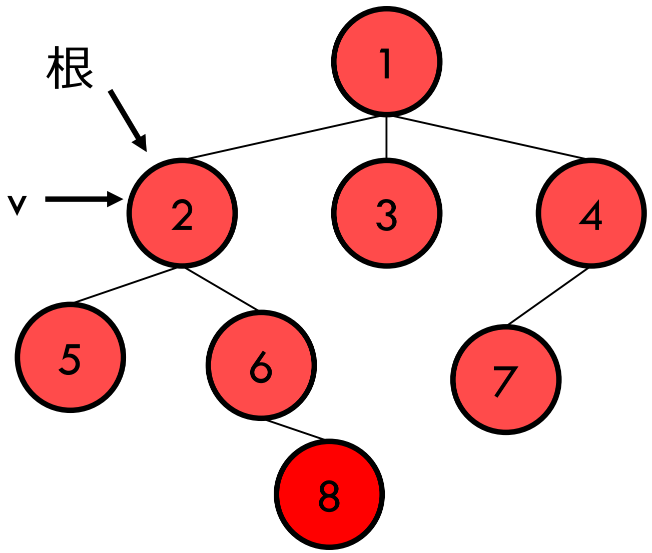
というのも、実際に根を変更する必要はなくて、現在の根の位置だけ持っておけばよい

- クエリ２'は根の位置に依存しないのでOK
- クエリ１'はそのままだとダメなので少し工夫が必要
  - →３つの場合分けが必要

クエリ１'の場合分け(1/3)

１つ目

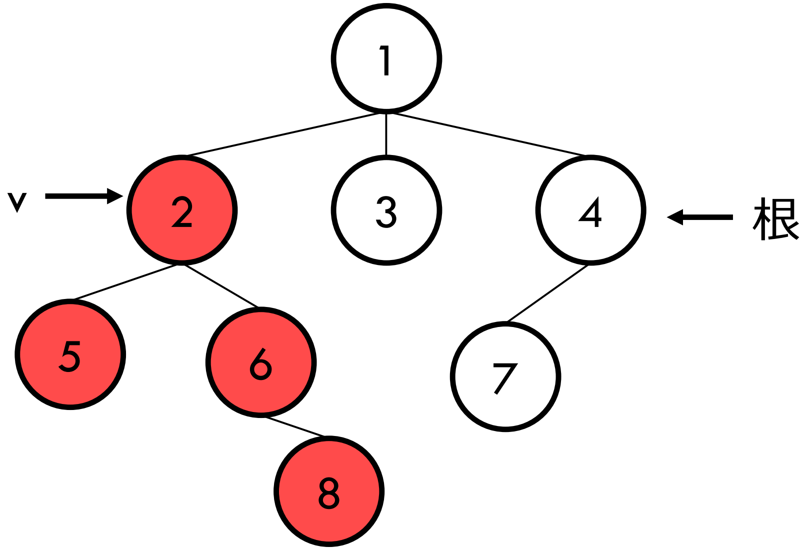
- ｖが根の場合、全体に足す



クエリ１'の場合分け(2/3)

２つ目

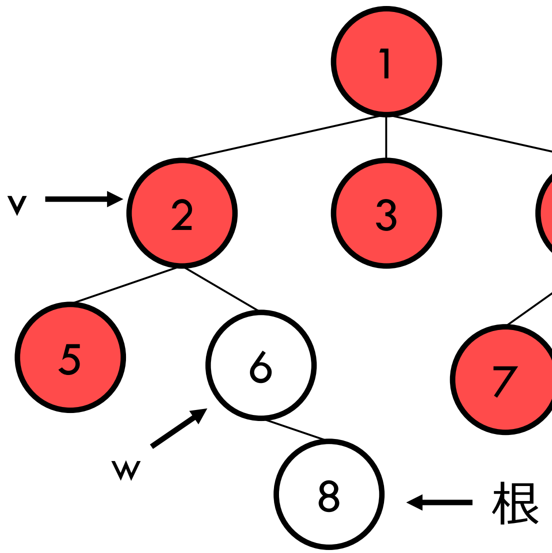
- 頂点１から見て、根が v の部分木に含まれていない場合、根が頂点１のときと同じ操作でよい
- 部分木にあるかどうかの判定はオイラーツアーでできる



クエリ１'の場合分け(3/3)

３つ目

- それ以外の場合、頂点 v から根の方向に１つ進んだ頂点を w とすると、全体の区間のうち w の部分木区間以外の区間に足せばよい
- w はダブリングなどで求まる



まとめ

クエリ１ ：$O(\log N \log X)$

クエリ２ ：$O(\log N \log X)$

クエリ３ ：$O(1)$

総計算量：$O(N \log N + Q \log N \log X)$

総メモリ：$O(N \log N + Q \log N)$