

Walking on Binary Tree

原案：someone_

問題概要

巨大な二分木を以下の条件に従って動く

i 回目の移動時, 以下のどちらかを選ぶ

- $S_i = L$ のとき左の子へ, $S_i = R$ のとき右の子へ動く
- 親へ動く (ただし親が存在する場合のみ)

u から v へ到達できるか判定し, 可能な場合必要な移動回数の最小値

Q クエリ, u, v は大きい

解法

$u \rightarrow \text{lca}(u, v) \rightarrow v$ と動く

- $u \rightarrow \text{lca}(u, v)$ は最短時間で登ってよい
- $\text{lca}(u, v) \rightarrow v$ は、進める場合と進めない場合がある
 - 進める場合 : 進んでよい
 - 進めない場合 : 逆側の子に下り, 次で戻る

ある最適な移動方法があったとき, 最適性を失わずこの動き方にできる

実装

1. 各 $i = 1, 2, \dots, N$ について「 i 以降いつ初めに左/右に下れるか」を
尺取り法などにより前計算

- このとき i と偶奇が一致するものしか見てはいけないことに注意

2. 各クエリでは愚直に移動しても $O(\log(u) + \log(v))$

- 前計算をサボると $\log 2$ つだが、定数倍のよい実装なら通るかも

$\text{lca}(u, v)$ は「 u と v が一致するまで大きい側を 2 で割る」が楽

統計情報

- AC team
 - 21 + ?
- First Acceptance
 - DaiMonge (00:29)