

Multi Ending Story

- 原案：稲葉
- 解答：宮村、山口
- 解説スライド：宮村

問題概要

- 二分木が与えられる。
 - あるノードでセーブする。
 - セーブしたノードに移動する。
 - 子ノードに移動する。
 - 葉ノードにいるとき、根へ移動する。
- の4つの操作ができる。全ての葉ノードを尋ねるときの子ノードへの移動回数を最小化せよ。
- ノード数 $V \leq 10^6$

解法

- 動的計画法を用いる。
- ノード v の下にある葉の数を $U[v]$ とする。
- ノード v の親でセーブして、セーブを一切用いずに以下の葉ノードを全て尋ねるのに必要なコストを $T[v]$ とおく。
- セーブ機能を適宜使いながら以下の葉ノードを全て尋ねるのに必要なコストを $S[v]$ とおく。

解法

- このとき、葉ノード以外で次が成り立つ。
- 今のノードを v 、右の子を r 、左の子を l 、親を p とする。
- $U[v] = U[r] + U[l]$
- $T[v] = T[r] + T[l] + U[v]$
- $S[v] = \min(T[l] + U[l] + 1 + S[r],$
 $S[l] + T[r] + U[r] + 1,$
 $S[l] + D[v] + S[r] + 2)$
- この関係式により $O(|V|)$ で計算可能。

コーナーケース

- 深さが大きい二分木が与えられたときスタックオーバーフローする可能性があるので注意。
 - 再帰を使わずに実装すればよい。
 - 1,2,...,N が自明なトポロジカル順序になっているので、再帰を使わない形への変換は簡単にできる。
- $T[i]$ は signed 32bit integer だとオーバーフローする可能性があることに注意。
 - 64bit integer で計算すれば足りる。