

H: Coins on a Tree

原案 : ei1333

問題文 : climpet

データセット : ei1333

解答 : climpet, ei1333, hos

解説 : ei1333

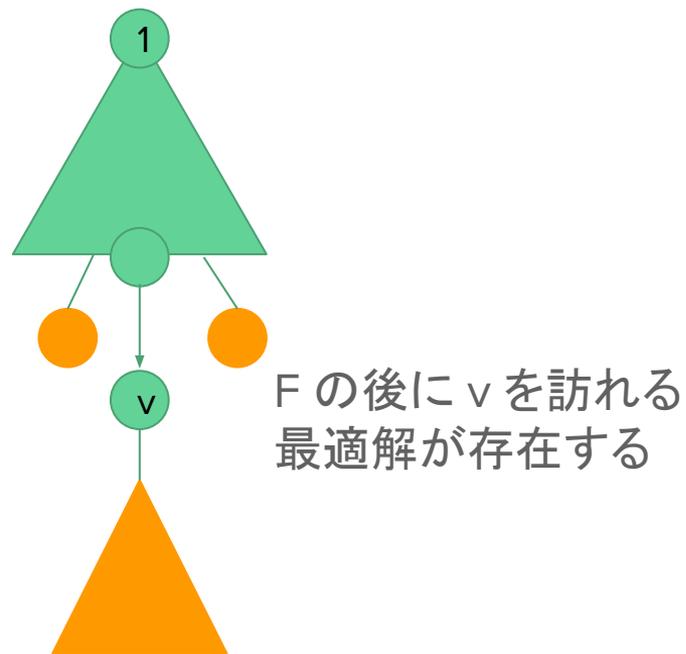
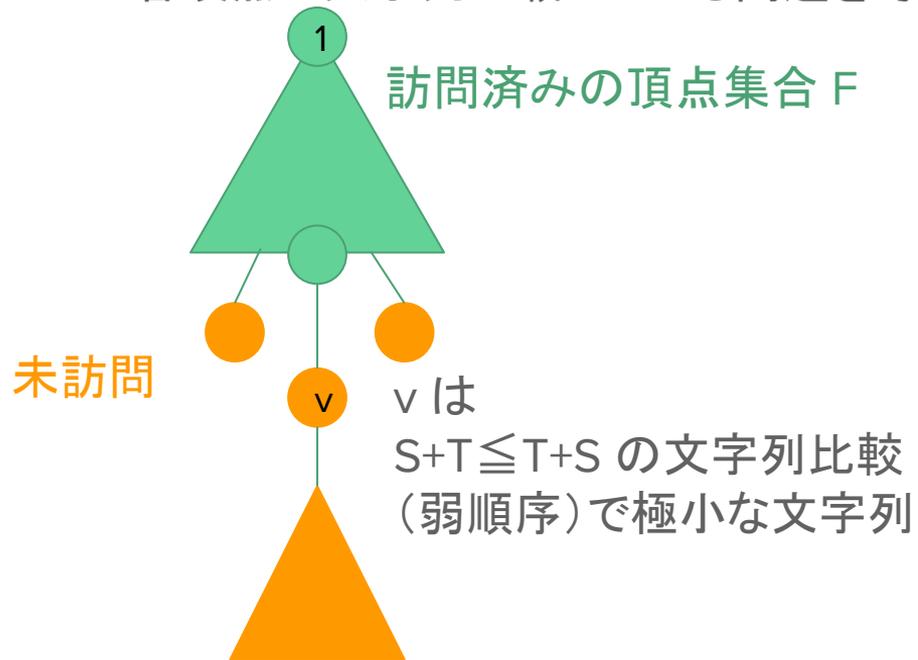
問題概要

- N ($2 \leq N \leq 2 \cdot 10^5$) 頂点の木が与えられる
 - 頂点 i ($2 \leq i \leq N$) の親は P_i
- 頂点 i ($1 \leq i \leq N$) には文字 C_i が書かれたコインが 1 枚置かれている

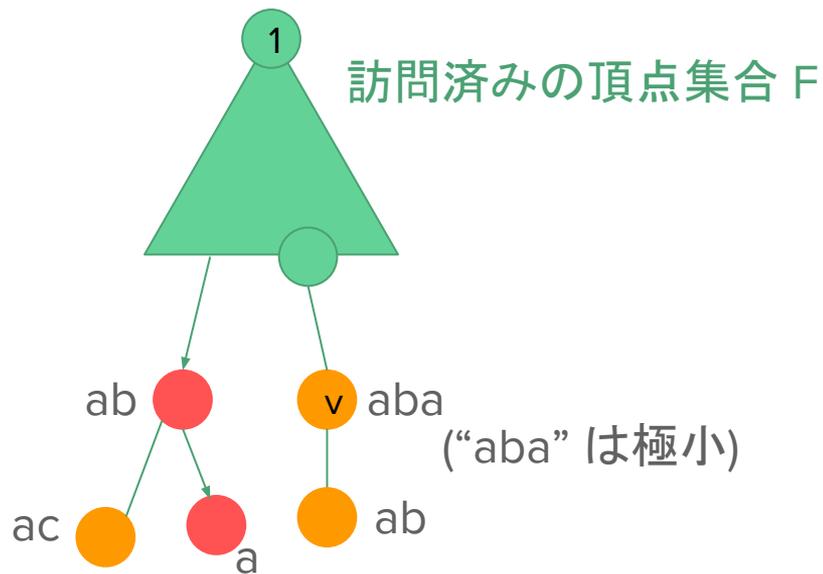
- 空文字列 S がある
- 頂点 1 を訪問し、木にコインが存在する間、隣接する頂点に移動することを繰り返す
- 頂点にコインがあれば回収し、 S の末尾にコインに書かれた文字を追加する
- S の可能な辞書順最小文字列を求めよ

考察 (thanks mtsd)

- 各頂点に文字列が載っている問題を考える



証明 (thanks mtsd)



- 頂点 1 と v の親を含み、 v を含まない任意の訪問順序を考える
- 次に v を訪れない場合、 v 上の文字列と同値関係になる経路を連続して訪れる必要がある
- このとき、 v を先に訪れても同じなので、 v を先に選ぶ最適解が存在する

解法

- 考察に基づくと、次の貪欲ができる
- 頂点が1つになるまで以下を繰り返す
 - a. $k = \operatorname{argmin}_{\{2 \leq i \leq N\}} C_i$ とする
 - ただし、比較は $C_i + C_j \leq C_j + C_i$
 - b. 頂点 P_k と k を縮約して、 $C_{\{P_k\}}, C_k$ を $C_{\{P_k\}} + C_k$ に置き換える
- 頂点同士の縮約は Union Find で効率的にできる
- 文字列に対する操作が大変そう

解法

- 以下の操作を効率的に行いたい
 - a. C_i ($2 \leq i \leq N$) のうち、辞書順最小文字列を求めて取り出す
 - b. 文字列同士を連結する
- 文字列に対応する Rolling Hash を持つとできる
 - a. set で管理する
文字列同士の比較は、対応する Rolling Hash 上で二分探索して LCP を求める
→ (LCP+1) 番目の文字を比較する
 - b. データ構造をマージする一般的なテクをする(次ページ)

Rolling Hash のマージテク

- 文字列 S に対応する (順方向の) Rolling Hash
 - $\text{hash}(S) = \sum_{0 \leq i < n} S_i b^i \pmod{p}$
- 先頭に文字 c を追加する
 - $\text{hash}(c + S) = (c b^{-1} + \text{hash}(S)) b \pmod{p}$
- 末尾に文字 c を追加する
 - $\text{hash}(S + c) = \text{hash}(S) + (c b^n) \pmod{p}$
- ハッシュを deque で管理し、先頭・末尾に追加した回数も同時に持つと $O(1)$ で各文字を追加可能 → 文字列 S, T のマージが $O(\min(|S|, |T|))$ ができる
- 全体の計算量 $O(N \log^2 N)$

ジャッジ解

- climpet(C++): 206 lines, 3788 B
- ei1333 (C++): 161 lines, 3552 B
- hos(C++): 235 lines, 7830 B

統計情報

- AC teams / Trying teams
 - 0 + 0 / 4
- First Acceptance
 - N/A