

JAG ICPC模擬国内予選2025

F: Anti-BCC

原案:hos

問題文:hint908

データセット:mtsd

解答:climpet, hos, kotatsugame, mtsd

解説:mtsd

問題概要

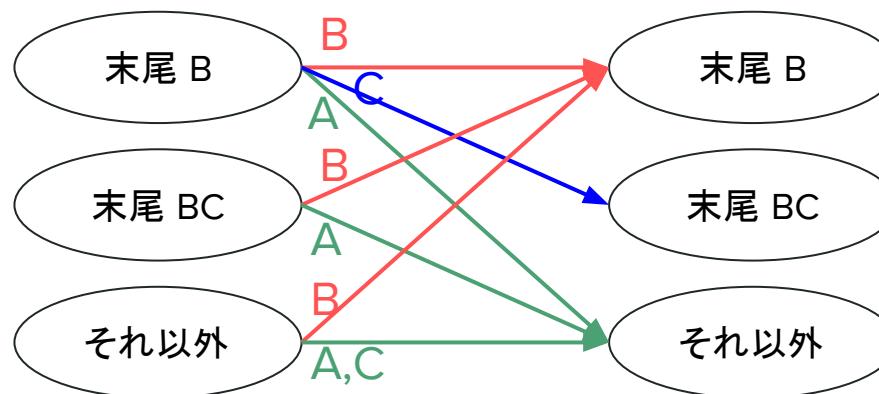
- A, B, C, ? からなる n 文字の文字列 S が与えられる
- 文字列 S の ? の部分を A, B, C のいずれかに変更して得られる文字列のうち、以下の条件を満たす辞書順最小のものを求めよ
(そのような文字列が存在しない場合はそれを報告せよ)
- 条件
 - A が a 個, B が b 個, C が c 個
 - 部分文字列として BCC を含まない
- 制約
 - $1 \leq n, a, b, c \leq 3,000$
 - $a+b+c = n$

TLE 解(存在判定) $\Theta(n^3)$

- 以下のようなテーブルを埋める動的計画法を考える
- $dp[len][cnta][cntb][\text{末尾B, 末尾 BC, それ以外}] = \text{True/False}$
 - 先頭からの長さが len
 - それまでに使った A の個数が $cnta$ 個
 - B の個数が $cntb$ 個
 - 末尾が B で終わる、BC で終わる、それ以外のパターン
 - BCCを含まないそのような文字列が存在するかを True/False で表現する

TLE 解(存在判定) $\Theta(n^3)$

- 以下のようなテーブルを埋める動的計画法を考える
- $dp[len][cnta][cntb][\text{末尾B, 末尾 BC, それ以外}] = \text{True/False}$
- 1文字追加する遷移は以下の通り



TLE 解(存在判定) $\Theta(n^3)$

- 以下のようなテーブルを埋める動的計画法を考える
- $dp[len][cnta][cntb][\text{末尾B, 末尾 BC, それ以外}] = \text{True/False}$
- $dp[len][a][b][*]$ で True のものが存在する場合、条件を満たす文字列は存在

想定解(存在判定) $\Theta(n^2)$

- validな文字列のCをAに変更してもvalidなままなので、Cを最も使ったパターンからC → Aの変更をすれば、AとCの個数は後から調整できる
- Bの使った個数に対して、使えるCの個数の最大値を記録すればよい
- つまり、以下のテーブルを動的計画法で埋めればよい
 - $dp[|len|][cntb][\text{末尾B}, \text{末尾BC}, \text{それ以外}] = (\text{使えるCの個数の最大値})$
- $dp[n][b][*]$ のいずれかがc以上であれば条件を満たす文字列は存在
 - ただし、元々の文字列に含まれるCの個数がcより多い場合NGであることに注意（サンプルにもあるコーナーケース）

想定解(復元パート) $\Theta(n^2)$

- 存在判定では先頭からの動的計画法を説明したが、
末尾からの dp テーブルを作ると復元しやすい
- $dp[i][cntb][B, BC, 空文字列]$ を以下の条件を満たす文字列が存在する
 C の個数の最大値とする
 - s の $[i, N)$ 文字目の ? を変更したものの中で
 - B を $cntb$ 個使った
 - 先頭に $\{B, BC, 空文字列\}$ を追加したときに BCC を含まない

想定解(復元パート) $\Theta(n^2)$

- $dp[i][cntb][B, BC, 空文字列]$ を使うことで、先頭から順に貪欲に辞書順最小の文字列を求めることが出来る
 - 初期状態を ($i = 0$, $cntb = 0$, 空文字列) として A, B, C の順に先頭に追加可能かを判定していくばよい

ジャッジ解

- climpet (C++) : 92 lines, 1950 B
- hos (C++) : 106 lines, 3138 B
- hos (Python) : 72 lines, 2566 B
- kotatsugame (C++) : 84 lines, 1717 B
- mtsd (C++) : 75 lines, 1937 B

統計情報

- AC teams / Trying teams
 - 12 / 31
- First Acceptance
 - 国内予選資格あり
 - DaiMongo (76 min)
 - Guest
 - Rubikun (58 min)