

E : Expression Sum

原案 : sapphire15

解説 : sapphire15

問題概要

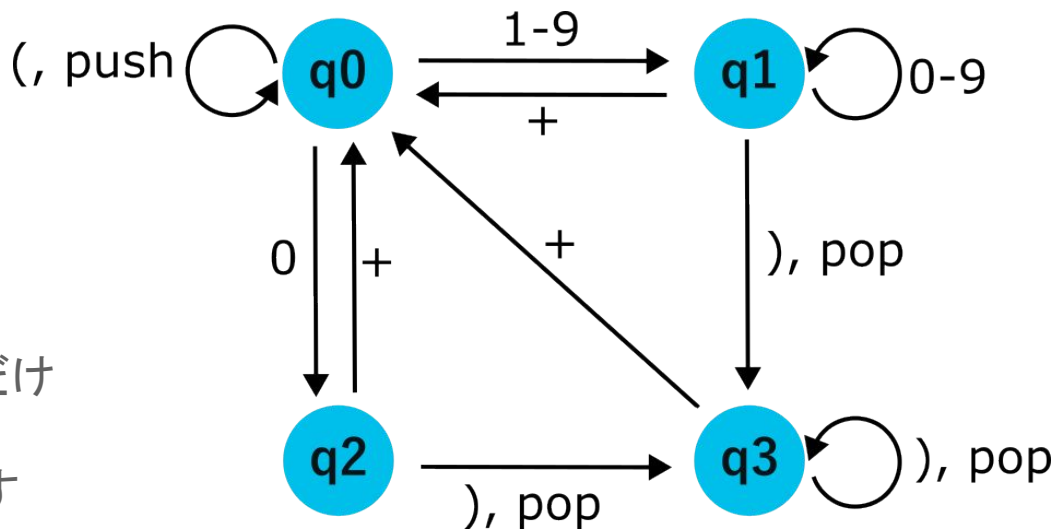
- 0123456789+()?からなる文字列 S が与えられる。
- S 中の?を0123456789+()のいずれかに置き換える。
- 置き換えた結果全てについて、置き換え後の文字列を計算式として評価したときの答えの総和を998244353で割ったあまりで求めよ。
 - 数式として正しくないものは無視する。
 - reading zeroはなし
 - 007 ✖
 - 7 ✔
- 制約
 - $|S| \leq 3000$

解法: valid expressionの識別

- 構文検査手法
 - 再帰降下構文解析
 - 区間DP(CKY法)
 - 3乗なので厳しい
 - プッシュダウンオートマトン
 - スタック付きのオートマトン
 - 今回はスタックに積む要素が1種類なのでDPで $O(N^2)$ とかできそう

解法: プッシュダウンオートマトン

- 以下のオートマトンの(状態, スタックの深さ)を管理すると valid expression の識別が可能



必要なときだけ
push, popを
書いています

解法: DPの中身

- ?がないとき

- (全体の総和, 今見ているnumberの値, 通り数(=1))^Tを保持
- 0-9の遷移

$$\begin{pmatrix} 1 & 9 & n \\ 0 & 10 & n \\ 0 & 0 & 1 \end{pmatrix}$$

を掛ける

+の遷移

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

- ?があるとき

- いろんな候補をまとめて遷移する
- オートマトンに合わせて上の行列を加算したものを掛ける

実装のときに注意すること

- BNFをよく読む
 - 問題ごとに微妙に違ったりする
 - 今回はreading-zeroがなし
- 定数倍がややきつい
 - 頑張ると $O(n^3)$ が速くてこうなりました $m(_ _)m$
 - 定数倍をサボりすぎるとTLEになるかも
 - ネストの深さを最小限にする
 - 1-9の遷移をまとめてする
 - 剰余をとるタイミングを工夫する
 - DPの状態数を抑える

統計情報

- Acceptances
 - 3 + 1 teams
- First Acceptance
 - AMATSUKAZE_world (123 min)