Additions

原案・問題文:not
データセット:tossy
解答:Darsein, not, tossy
解説:tossy

問題概要

0〜9と+からなる文字列 S と自然数 N が与えられる。($|S| < 10^3$, $N < 10^9$)

足し算のみからなる正しい数式に書き換え、計算結果を N 以下にしたい。

（ただし、単項演算子や leading zero は認めない。）

このとき、書き換える文字数の最小個数を求めよ。
（不可能な場合は-1を出力。）

例

+	1	2	3

0	+	2	3

2箇所の変更で23

+	1	2	3

0	+	1	0

4箇所の変更で10

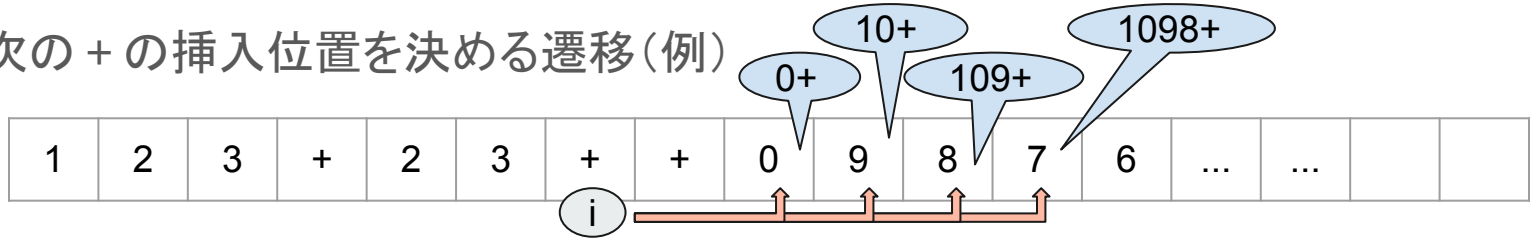
考察

- 計算結果として得たい値は高々10^9なのに対して、文字列長は10^3もあるため、一般には + がたくさん挿入される。
- dp[i][j] = (先頭から i 文字目までで j 文字置換し、+で終わる正しい式にしたときの、式の評価値としてとりうる最小値)
を前から更新していくDPを考える。

方針

dp[i][j] = (先頭から *i* 文字目までで *j* 文字置換し、+で終わる正しい式にしたときの、式の評価値としてとりうる最小値)

次の + の挿入位置を決める遷移（例）



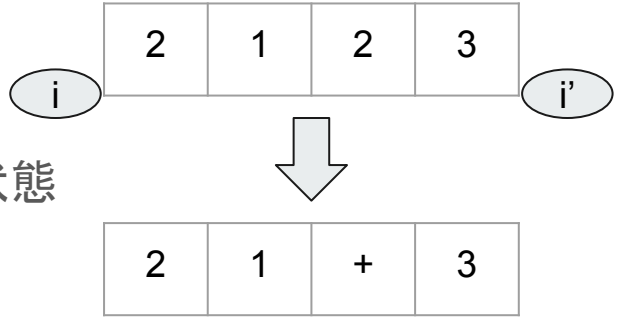
- *i* の増加分は $\log_{10} N$ 程度までで十分。（それ以上増やしても、式の評価値がNを超えてしまう）
- + は 0/1 に置換する。（leading zero になるときは1を入れる）

DPの遷移（方針1 - $O(|S|^2 \log^2 N)$）

- 遷移の開始点 i と，次の遷移先 i' を決める。
- これから書き換える文字数 k を決める。
- i と i' の間にある文字は次のような貪欲の書き換えで、遷移先に渡す値が決まる:
  - `+` を 0 にする。（leading zeroになるときは 1 ）
  - 全ての `+` を書き換えられなかったら，その遷移は無理。
  - 全ての `+` を書き換えたら、残りを `1000...` を作るようにする。
- 遷移や計算の順番を整理すれば、上記計算量でとける。

DPの遷移（別解 - O(|S|² logN)）

- 桁数3以上の数値をつくる遷移のとき、ある数字を 0/1 に書き換えるのは意味がない。
  - どこかを `+` に置換すれば桁数が半分程度になるので、大きい桁数では、数字を数字に置換すべきではない。
- 遷移中は，`+` を0/1に置き換えるもののみ考えればよくなり、遷移を削れる。
  - 桁数3以上を作る遷移は、一つ小さい桁数の状態から桁数を伸ばす場合のみ考える。
  - 桁数1,2だけコーナーを考慮すればよい。



ジャッジ解

- **$O(|S|^2 \log^2 N)$**
  - not : 101行
  - Darsein : 59行
- **$O(|S|^2 \log N)$**
  - tossy ー 97行

結果

- First AC (online)
  - ei13333333333（94:12）
- First AC (onsite)
  - narianZ（119:32）

- 方針はそこまで難しくないですが、遷移の細かい条件を網羅できずにWAとなるチームが目立ちました。