ACM-ICPC OB/OG Spring Contest 2014
Problem D - LR

原案 : 楠本
解答 : 山口、大槻、保坂、森
解説 : 大槻

問題概要

- 以下で定義される関数 L, R がある
  - L(x, y) = x, R(x, y) = y
- “R?????,2?)” のような文字列が入力として与えられる
  - 各文字は "LR(),0123456789?" のいずれか
- ‘?’ を “LR(),0123456789” のいずれかに置換して，関数の数式にしたい
- そのような数式のうち，数式の値の最大値を求めよ
  - 上の例は例えば “R(1111,29)” として R(1111,29) = 29 が最大

サンプル

- “?????,??,???”
  - R(L(1,1),99) = 99 が最大
- “????,??,???”
  - invalid
- “L?11?????????????????????????????????????”
  - L(19999999999999999999999999999999999,1) = 19999999999999999999999999999999999 が最大
  - **多倍長整数**も登場！

基本的な考察

- 関数の数式は入れ子構造になっている

  R(L(R(3, 4), 5), L(5, 6))

- 関数の数式は L(,)のような形

区間DPで解けそう

想定解法 --- 区間DP

i　　　j-1

"R(??3,?　…　?(??8??　…　1?)?"

- dp[i][j] := i文字目からj-1文字目までの部分についての最大値
- 部分文字列全体は，L(　,　)か R(　,　)

カンマ位置(＝kとする)で場合わけ

- i～j-1全体を数値numにできれば dp[i][j] = max(dp[i][j], num)
- i+1，j-1文字目を（　)にできるとき
  - i 文字目をLにできるとき dp[i][j] = max(dp[i][j], dp[i+2][k])
  - i 文字目をRにできるとき dp[i][j] = max(dp[i][j], dp[k+1][j-1])

注意点 --- 多倍長整数

- 多倍長整数が必要
- とは言え大小比較さえできればOK！
  - 文字列のままで実現できる！
- A ＞ B とは
  - 「Aの文字数 ＞ Bの文字数」 or
  - 「Aの文字数 ＝ Bの文字数」で「A ＞ B（辞書順）」

教訓 --- 見た目にだまされない

- 見た目ヤバそうな問題だが（本セット内では）そこまで難しくない
- だが，最初の1時間は正解者ゼロであった
- 序盤は A の後に E，H に挑むチームが多かったが，Dより嵌りやすく苦戦したチームも多数
- 順位表を過度に信じ過ぎず，解けそうな問題を着実に見極めよう

結果

- First AC
  - wakaba
  - 01:02:30

- Accepted / Submissions
  - 21 / 35（60%）