



Golf

JAG 模擬国内予選 2014

原案：村主

解答：井上・須藤・保坂・森・山口

解説：保坂



問題概要

- 加・減・乗・整数除・冪乗・括弧の式で N を作るときの最短の長さは？
 - $0 \leq N \leq 2^{36} - 1$
 - 途中の値も 0 以上 $2^{36} - 1$ 以下でなければならない

Sample Input

- $117649 = 7^6$
- 1 (短くできない)
- $125000 = 50^3$
- $1610612736 = 8^9 * 12$
- 68719476636 (短くできない)
 - $2^{36} - 10$ はダメ

考察

- 冪乗が絡まないと短くならない！
 - $(m \text{桁の数}) \pm (n \text{桁の数})$ は $\max(m, n) + 1$ 桁以下
 - $(m \text{桁の数}) \times (n \text{桁の数})$ は $m + n$ 桁以下
 - $\lfloor (m \text{桁の数}) \div (n \text{桁の数}) \rfloor$ は m 桁以下
 - どれも, 演算を使わずそのまま書いたほうが短い

考察

- 冪乗 a^b で最低 3 文字使う
- 10 文字以下にしなければいけない
 - $2^{36} - 1$ は 11 桁
- ということはあと 3 演算しかできない！
 - 演算 1 回増やすと 2 文字は使ってしまうから

考察

- 「使える」冪乗 a^b は？
 - $a < 2$ や $b < 2$ は要らない
 - $a^b < 2^{36}$ なので $a < 2^{18}$ や $b < 36$ がわかる
- (式)^b とか $a^{(式)}$ とかも要らない
 - "(式)" を書くのに 5 文字も使ってしまう
- 短くできる冪乗を列挙してしまえる
 - 300000 個もありません

解法

- 数式のパターンを全部調べる！
- 以下冪乗で作るものを大文字で書きます
 - 大文字で書いたら長さ 3 以上
- $P+a$ とか, $P*a$ とか, $P*a-b$ とか, $P+Q$ とか, ……
 - 一方 $P+a+b$ とか $P/a/b$ とかは要らない
 - 定数部分まとめたほうが短いから

解法

- 数式のパターンを全部調べる！
 1. 数式のパターンを決めた後どうするの？
 2. 数式のパターンをどうやって挙げるの？

解法

1. 数式のパターンを決めた後どうするの？
 - 愚直な全探索は時間がかかりすぎる
 - 例えば $P * a + b$ というパターン
 - 作る数 N がわかっているなら, P と a を決めれば b は $b = N - P \times a$ で求まる！
 - P と a を全探索 → b を求める → 条件チェック

解法

1. 数式のパターンを決めた後どうするの？

– 他のパターンも, 1 箇所以外を決めて逆算すればよい

- $P+Q$ とか $P*Q$ とかがあるので冪乗の値から長さを表引きできるようにしておく

– 十分速いのはなぜ??

- 最大 10 文字のうち 1 箇所以外を決めるので, 残りは最大 8 文字分, ということは高々 10^8 通り
 - 冪乗が絡む必要があることから実際はもっと減る

解法

1. 数式のパターンを決めた後どうするの？

– 逆算の方法

- 加算・減算は簡単
- $x \times y = N \implies y = \lfloor N \div x \rfloor$
- $\lfloor x \div y \rfloor = N \implies y = \lfloor x \div (N + 1) \rfloor + 1$
 - 証明： $yN \leq x < y(N + 1)$ を変形
 - 候補がこの 1 通りになるのでこれを試す
 - うっかり候補を $y = \lfloor x \div N \rfloor$ とやってしまっても本問では間違えるケースは存在しません

解法

2. 数式のパターンをどうやって挙げるの？

① 手作業で列挙

- メリット：PC を他の問題に回せる
- デメリット：ミスしやすい

② プログラムを書く

- メリット：コーディングで場合分けが減る？
- デメリット：難しい？

③ 適当なプログラム＋手作業

- 式の形をとりあえず適当に列挙してしまっても、不要なパターンを手作業で除く

数式のパターン

- 例：項数ごとに列挙してみる

1. $a, P, P*a, P/a, P*a/b, P/a*b, P*a/b*c, P/a*b/c, P*Q, P/Q, P*Q*a, P*Q/a, P/Q*a, P/Q/a, (P+a)*b, (P+a)/b, (P-a)*b, (P-a)/b$

2. $P+a, P-a, P*a+b, P*a-b, P/a+b, P/a-b, P*a/b+c, P*a/b-c, P/a*b+c, P/a*b-c, P*Q+a, P*Q-a, P/Q+c, P/Q-c, P+Q, P-Q, P*a+Q, P*a-Q, Q-P*a, P/a+Q, P/a-Q, Q-P/a$

3. $P+Q+a, P+Q-a, P-Q+a, P-Q-a$

- 定数から引いたり定数を割ったり、というタイプで損になるパターンもここで除いています
- 実は不要とわかるものもいくつか入っています

実装

- がんばってください
- オーバーフローに注意
 - $x \times y$ を計算する前に $x > 2^{36}/y$ かどうか判定しましょう
- 0 の桁数を 0 と言わないように注意
- 途中式の値が範囲内に収まっていることをちゃんとチェックする
 - 解の候補を作るたびにチェック関数に投げる

別解

- 10 文字以下では P^*a+Q^*b のようなパターンが不要で、「冪乗から始めて 1 個ずつ追加していく」ですべてのパターンが作れることがわかる
 - 別の言い方をすると、構文木で leaf 以外の node が必ず leaf を子に持つということです
- 8 文字までで作れるものを DP 的にリストアップしていくと場合分けがほぼ不要
 - 1 項か 2 項以上かで状態を分ける

見落とししやすいパターン

- 解答を作成したスタッフ全員が最初の実装では何かしら見落とししたのでそういうものを紹介します

見落とししやすいパターン

- $P+Q+a, P+Q-a, P-Q+a, P-Q-a$
 - 例 : $30523342925 = 5^{15}+7^8-1$
 - 3 項使うことはあります
- $P*Q/a$
 - 例 : $10942057291 = 7^5*5^9/4$
 - 定数以外も掛け算しうるしその後もあります
 - $P*Q*a$ も一見同様に必要そうですが $P*a$ のパターンでカバーできてしまうようです

見落としやすいパターン

- $P*a/b*c, P/a*b/c$
 - 例 : $21624389644 = 15^9/8*9/2$
 - 切り下げを利用できるので掛け算と割り算はまとめられません
- $Q-P*a, Q-P/a$
 - 例 : $29742737147 = 5^{15}-2*9^9$
 - 引き算は逆向きにも使えることを忘れがち

見落としやすいパターン

- $(P-a)*b$
 - 例： $68719476664 = (2^{33}-9)*8$
 - 括弧は基本的には展開すれば同じか短くなるのですが……
 - 展開してしまうと 2^{36} かもしれません
 - 余計な考察をしないで実装せよ, という教訓?
 - 11 文字以上だともっと普通に括弧が要ります

ジャッジ情報

- 井上 : 169 行 4144 B (C++)
- 須藤 : 188 行 4392 B (C++)
- 保坂 : 360 行 16903 B (C++)
- 山口 : 535 行 26881 B (Java)
 - 保坂解の翻訳
- 保坂 (別解) : 193 行 4529 B (C++)
- 森 : 290 行 8119 B (C++)

結果

- Accepted / Trying Teams / Submission
– 0 / 1 / 1
- Trying Teams
 - !#\$%&()*+-. /: ; < = > ? @ [¥] ^ _ ` { | } ~
 - デバッグがあと少しのところだったようです