Domača naloga #1: cela števila
Priprave na računalniške olimpijade 2018/19

Tomaž Hočevar
tomaz.hocevar@fri.uni-lj.si

A. Accounting

Omejitve v nalogi so precej nizke ($|A|, |B| \leq 1000$). Tako lahko preverimo vse možne rešitve X od -1000 do 1000. Ker tudi potenca n ne preseže 10, ne potrebujemo hitrega potenciranja. Vsako potencialno rešitev X zmnožimo n-krat in preverimo, ali ustreza enačbi. Pri tem moramo paziti, da produkt ne preseže velikosti podatkovnega tipa.

Seveda obstaja tudi učinkovitejša rešitev, ki izračuna n-ti koren od B/A. Paziti pa moramo na robne primere, kot je npr. $A = 0$. Iščemo najenostavnejšo rešitev, ki je dovolj dobra, saj je tam verjetnost napak najmanjša. Zato je v tem primeru prvi pristop boljša izbira, kot ste nekateri izkusili na lastni koži.

B. Primes or Palindromes?

Število praštevil do n je približno $n/\log n$, število palindromskih števil pa približno $\sqrt{n}$. Iz tega lahko izračunate, ali pa preizkusite, da odgovor nikoli ne bo večji od 10^7.

Pripravite si lahko dve tabeli velikosti 10^7, kjer na i-tem mestu shranite podatek, ali je število i praštevilo oz. ali je palindrom. Tabelo praštevil izračunate z Eratostenovim rešetom, za izračun tabele palindromov pa morate obrniti vsako število. Lahko ga pretvorite v niz in obrnete ali pa to narediti kar z operacijami nad celimi števili - z ostankom pri deljenju z 10 izločite zadnjo števko ($d = x \mod 10$) in jo dodate obrnjenemu številu ($y = 10y + d$).

Sedaj se samo sprehodite čez obe tabeli in si zapomnite, kdaj je bilo razmerje med številom praštevil in številom palindromskih števil nazadnje pravilno. Ne smete pa preiskovanja zaključiti, čim se razmerje podre; morda se par števil kasneje spet popravi.

C. Reducing Fractions

Na prafaktorje razcepimo vsa števila $a_1, \ldots, a_n$ in $b_1, \ldots, b_m$. Za vsak prafaktor si zapomnimo, kolikokrat se pojavi v števcu in kolikokrat v imenovalcu. Če je razlika pozitivna, se bo potenca tega prafaktorja pojavila v števcu rezultata, sicer pa v imenovalcu. Enostavno? Ne povsem.

Prva ovira so omejitve. Razcepiti moramo 10^5 števil velikosti 10^7. Z iskanjem prafaktorjev do korena vsakega števila bomo prepočasni. Tu nam pride prav razširjeno Eratostenovo rešeto, kjer si za vsako število x shranimo posamezen prafaktor, zaradi katerega x ni praštevilo. Tako lahko po izvedbi rešeta vsako število razcepimo v $O(\log x)$ operacijah.

Na drugi oviro pa naletimo ob izpisu, ki ima enake omejitve kot vhodni podatki. Seznam praštevilskih faktorjev v števcu in imenovalcu moramo združiti v števila, ki niso večja od 10^7 in jih skupaj ni več kot 10^5. Razne požrešne strategije množenja prafaktorjev, dokler ne presežejo 10^7, se ne obnesejo. V bistvu gre za precej težek problem, kako optimalno združevati te prafaktorje. Na srečo pa si lahko pomagamo z vhodnimi podatki, kjer so ti prafaktorji (poleg še nekaterih drugih, ki smo jih okrajšali) že pravilno združeni. Če vhodnim podatkom (vrednostim a_i in b_i) odstranjujemo prafaktorje, namesto da bi jih sestavljali na novo, imamo zagotovljeno pravilno obliko izpisa.