Domača naloga #4: izčrpno preverjanje
Priprave na računalniške olimpijade 2018/19

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

A. Three Sons

Dimenzije polja ($n, m \leq 50$) so dovolj majhne, da lahko preverimo vse možne pare navpičnih delitvenih črt. Za vsako delitev seštejemo celice v vseh treh območjih in jih primerjamo z zahtevano distribucijo (tri izračunane vsote uredimo in urejen seznam primerjamo z urejeno distribucijo). Da se izognemo napakam pri kopiranju kode še za vodoravne delitvene črte, tabelo transponiramo in ponovimo postopek. Glede na omejitve lahko celice v vsakem območju seštevamo vedno znova. Če bi hoteli biti bolj učinkoviti, si vnaprej pripravimo predponske vsote, s katerimi izračunamo vsoto v določenem območju v konstantnem času.

B. Lucky Numbers (easy)

Srečna števila so sestavljena samo iz števk 4 in 7, torej jih ni prav veliko. Število k-mestnih srečnih števil je 2^k. Največji možen odgovor je pri $n = 10^9$ in sicer 4444477777. Generiramo lahko torej vse srečna števila (ki so največ 10-mestna) in za vsako od njih preverimo, ali je zelo srečno (ima enako število štiric in sedmic). Izmed vseh najdenih zelo srečnih števil izberemo najmanjše, ki presega podano mejo n. Srečna števila lahko generiramo rekurzivno. Parametri rekurzivne funkcije bi bili npr. zgrajeno število x, število štiric n_4 in število sedmic n_7. Številu x lahko dodamo števko $d = \{4, 7\}$ in nadaljujemo generiranje z $10x + d$.

C. Polycarpus the Safecracker

Nalogo lahko rešimo z zelo učinkovitim rekurzivnim generiranjem vseh tabel. Po vrsticah vpisujemo nova praštevila, pri tem pa moramo paziti, da se predpone teh praštevil ujemajo z že vnesenimi. V ta namen nam bo prav prišel organiziran seznam praštevil. Za velikost tabele n, velikost predpone k in vrednost predpone x, si lahko shranimo seznam praštevil, ki temu ustrezajo (npr. $f(5, 3, 207) = 20707, 20717, 20719, ...$). Pri tem moramo paziti na to, da imajo lahko praštevila v tabeli tudi vodilne ničle. Prav nam pride tudi vnaprej pripravljeno razbitje praštevil na števke, saj je ponavljajoče izločanje števk s pretvarjanjem v nize ali z računanjem ostankov pri deljenju z 10 zamudna operacija.

V alternativni rešitvi lahko opazimo, da so vrednosti na diagonali neodvisne med seboj. Izbira ene izmed njih ne vpliva na to, kaj lahko izberemo na preostalih mestih. Če poleg podane prve vrstice izberemo vrednosti v vseh celicah nad diagonalo (teh možnosti je 10^6, ker imamo 6 prostih celic), nam to definira celotno tabelo z izjemo celic na diagonali. Če sedaj ugotovimo, da lahko v drugo diagonalno celico vpišemo a_2 števk, v tretjo a_3, itd., obstaja $a_2 \cdot a_3 \cdot a_4 \cdot a_5$ možnih rešitev, ki jih lahko prištejemo, ne da bi vse generirali.