Univerzitetni programerski maraton 2020
2. kolo — rešitve nalog

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

7. maj 2020

256 odtenkov sive

Naloga je precej enostavna. Iz prebranih števil moramo izluščiti relevantne bite, da dobimo vrednosti rdeče, zelene in modre. To lahko počnemo z bitnimi operacijami, še lažje pa z deljenjem in ostanki pri deljenju z 256. Nato samo izpišemo zaokroženo uteženo vsoto, kot jo definira naloga.

Hrošč

Hrošč se v vsaki potezi premakne z lihega na sodo mesto ali obratno. V dveh potezah ostane na mestu ali pa se premakne za dve mesti levo ali desno. Predpostavimo, da se hrošč nahaja na sodem mestu in ga poiščimo. Preverimo škatlo 0 in če ga ni, se mora po premiku nahajati nekje na $[1, 3, 5, \ldots]$. Nato preverimo škatlo 1 in ob neuspehu ugotovimo, da mora biti nekje na $[2, 4, \ldots]$. Z isto logiko preverimo vse škatle do N. Če ga nismo našli, je bila naša predpostavka o parnosti začetne škatle napačna. Izračunamo, v škatli katere parnosti se mora trenutno nahajati in ponovimo postopek. Enostavna rešitev, ki doseže isto je $[0, 1, \ldots, n-1] + [n-1, n-2, \ldots, 0]$.

Sortirna postaja

Pred združevanjem vagonov na izhodni tir, morajo biti vsakem sortirnem tiru številke vagonov urejene. To je potreben in zadosten pogoj, da jih lahko združimo v urejeno zaporedje. Pri razporejanju vagonov z vhodnega tira na sortirne, je torej pomembna samo številka zadnjega vagona na posameznem tiru. Če so vse številke večje od številke novega vagona x, ga moramo postaviti na nov tir. Sicer imamo na izbiro vse tire, ki imajo na koncu vagone s številkami manjšimi od x. Če ta nabor kandidatov ni prazen, ni smiselno odpreti novega tira. Med kandidati (manjšimi od x) lahko s požrešno strategijo vedno izberemo največjega, recimo y. Intuitivno naredimo s tem najmanj škode - vagoni s številkami $y - x$ bodo morali na nek drug tir. Formalno lahko to dokažemo z argumentom zamenjave, kjer požrešna poteza ne more poslabšati optimalne rešitve, ampak ostane enaka ali jo celo izboljša. Izkaže se, da isto dosežemo s strategijo vstavljanja na prvi tir, ki to omogoča. Posledično so tiri urejeni od tistih z večjimi proti tistim z manjšimi številkami vagonov na koncu.

Jenga

V nalogi je natančno definiran pogoj za stabilnost stolpa. Omejitve vhodnih podatkov pa so dovolj nizke, da lahko za vsako odstranjeno ploščico ločeno ugotavljamo stabilnost spremenjenega stolpa. Stolp je stabilen, če je težišče kvadrov nad vsako delitveno ravnino med nivoji podprto s kvadri neposredno pod ravnino. Delitvene ravnine preverjamo od zgoraj navzdol in sproti popravljamo težišče kvadrov. Posamezen primer lahko torej rešimo v $O(H^2)$.

Naredimo par poenostavitev, ki nam bodo olajšale reševanje. Natančna masa kvadrov in njihove dimenzije so nepomembne. Za izračun težišča potrebujemo povprečno x-koordinato in y-koordinato težišča kvadrov v zgornjih nivojih. Če postavimo izhodišče koordinatnega sistema na sredino stolpa, vsak nivo vpliva samo na spremembo povprečja ene ali druge koordinate. Namesto iskanja treh podpornih točk, ki definirajo trikotnik, lahko razmišljamo o konveksni ovojnici vseh podpornih točk (lahko jo razbijemo na trikotnike). Vedno je problematična podpora samo v eni smeri, saj se v nasprotno smer raztezajo kvadri po celotni dimenziji. Potrebujemo samo najmanjše in največje težišče podpornega kvadra v tej smeri, ki ju za potrebe ugotavljanja podpore razširimo še za pol enote. Če je povprečno težišče izven teh dveh meja, stolp ni stabilen.

Protesti

V nalogi gre za precej očiten primer vozliščnega pokritja v dvodelnih grafih. Komplikacija z ugotavljanjem lokacij protestov iz nepopolnih podatkov je dokaj enostavna ovira do izgradnje dvodelnega grafa. Prvo množico vozlišč predstavljajo obvoznice, drugo pa vpadnice. Vozlišči sta povezani, če na njunem križišču poteka protest. Iščemo najmanjšo množico vozlišč (cest), s katerimi pokrijemo vse povezave (protesti).

Konigov izrek pravi, da je velikost največjega prirejanja v dvodelnem grafu enaka najmanjšemu vozliščnemu pokritju. Za iskanje prirejanja obstaja enostaven algoritem s časovno zahtevnostjo $O(VE)$, kjer je V število vozlišč, E pa število povezav. Dokaz izreka nam pove tudi, kako lahko iz največjega prirejanja enostavno skonstruiramo najmanjše vozliščno pokritje.

LCM seznami

Produkt števil je enak najmanjšemu skupnemu večkratniku, če so si vsa števila med seboj tuja. Koristno bi bilo razcepiti vseh N števil (ki ne presežejo A) na prafaktorje, vendar je faktorizacija v $O(N\sqrt{A})$ prepočasna. Prilagodimo lahko Eratostenovo rešeto s časovno zahtevnostjo $O(A \log \log A)$, kjer namesto značke, ali je neko število praštevilo ali ne, hranimo najmanjši praštevilski faktor (zaradi katerega ni praštevilo). Potem lahko vsako število faktoriziramo v času, ki je odvisen samo od števila njegovih prafaktorjev, kar ne preseže $O(\log N)$.

Za vsak indeks i lahko izračunamo najmanjši indeks l_i, da so si vsa števila v seznamu $S[l_i..i]$ tuja. Za vsako praštevilo moramo hraniti njegovo zadnjo ponovitev, da lahko poiščemo najbolj desnega izmed prafaktorjev števila s_i. Tako dobimo seznam različno dolgih intervalov. Za poizvedbo (a_i, b_i) pridejo v poštev vsi intervali z desnim robom med a_i in b_i. Iščemo vsoto njihovih dolžin, pri čemer ne upoštevamo delov, ki se raztezajo čez a_i.

Očitno naraščajo desni robovi teh intervalov, uporabna lastnost pa je, da naraščajo tudi levi. Tako lahko z bisekcijo poiščemo prvi interval, ki je v celoti znotraj iskanega območja (in mu ni treba odrezati kakšnega dela), recimo mu m. Sešteti moramo dolžine intervalov $m, \ldots, b_i$ in pri a_i odrezanih intervalov $a_i, \ldots m-1$. To lahko počnemo učinkovito s predponskimi vsotami - vsoti desnih robov odštejemo vsoto levih.