Univerzitetni programerski maraton 2019
1. kolo — rešitve nalog

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

24. april 2019

Kuponi za pice

Prijateljem se splača naročati čim dražje pice, ker jih zanima privarčevan denar na račun kuponov in ne skupno plačilo. Kupone bo nato unovčilo k prijateljev z najcenejšimi picami. Poiščemo torej največjo vrednost v vsaki vrstici, te vrednosti uredimo in seštejemo najmanjših k.

Gozdovi

Preštetі moramo povezane komponente v grafu, ki je definiran s tabelo rasti dreves. Vsaka celica tabele ustreza vozlišču v grafu, sosednja vozlišča pa so tiste izmed osmih sosednih celic, ki vsebujejo isto vrsto dreves. Komponente lahko poiščemo s preiskovanjem v širino ali v globino. Pri slednjem bodite pozorni na število rekurzivnih klicev.

Barvanje drevesa

Če imamo eno samo vozlišče, potrebujemo zgolj eno barvo. Če ima drevo obliko poti (brez vejitev), zadostujeta dve barvi. Sicer potrebujemo tri barve. Vozlišča barvamo po nivojih od korena proti listom in od leve proti desni. Ko želimo pobarvati neko vozlišče, ima to največ dve bližnji vozlišči, ki sta že pobarvani. To sta starš in nazadnje pobarvano vozlišče na trenutnem nivoju. Zagotovo bo ena izmed treh barv še prosta. Pri implementaciji rešitve nam pride prav seznam vozlišč po nivojih, ki si ga lahko pripravimo v naprej.

Ocenjevanje univerz

Velja $x + y = 1$, zato lahko formulo za kvaliteto prepišemo v $k = ax + b(1 - x)$. Ker želimo, da je naša univerza najboljša, dobimo sistem neenakosti

$$\begin{aligned} a_1x + b_1(1 - x) &\geq a_ix + b_i(1 - x) \\ x((a_1 - b_1) - (a_i - b_i)) &\geq b_i - b_1. \end{aligned}$$

Glede na faktorja $f = (a_1 - b_1) - (a_i - b_i)$ in $g = b_i - b_1$ obravnavamo več primerov. Dobimo lahko nemogočo situacijo, zgornjo mejo za x ali pa spodnjo mejo za x. Če po obravnavi vseh neenakosti interval možnih vrednosti za x ni prazen, lahko izberemo poljubno vrednost z intervala.

Hosoyev indeks

Naj bo $f(n)$ iskano število prirejanj v helicenu velikosti n. "Zadnja povezava" v n-tem benzenu je lahko del prirejanja ali pa ne. Če je, sosednji povezavi ne moreta biti del prirejanja in dobimo "deformiran" podproblem $f_{1,1}(n)$, kjer ima n-ti benzen samo po eno povezavo na vsaki strani namesto celega obroča. Če ta povezava ni del prirejanja, pa imamo opravka s podproblemom $f_{2,2}(n)$. Tako dobimo sistem enačb:

$$f(n) = f_{2,2}(n) + f_{1,1}(n)$$
$$f_{2,2}(n) = f_{1,2}(n) + f_{0,2}(n)$$
$$\dots$$

Ker so problemi vedno manjši, lahko enačbe razpišemo do te mere, da je v vsaki enačbi funkcija z argumentom n odvisna samo od funkcij z argumentom $n-1$:

$$f(n) = 5f(n-1) + 3f_{0,3}(n-1) + f_{1,2}(n-1) + 2f_{2,2}(n-1)$$
$$f_{0,3}(n) = 3f(n-1) + 2f_{1,2}(n-1)$$
$$\dots$$

Tako dobimo linearen sistem rekurenčnih enačb, ki ga lahko zapišemo v matrični obliki $v_n = A^{n-1}v_1, \quad v_i = [f(i), f_{0,3}(i), ...]^T$. Poznamo A in v_1, izračunati pa moramo v_n. Matrika A je dimenzij 4×4 in jo lahko potenciramo s kvadriranjem v $O(\log n)$.

Premikanje pohištva

Ker optimiziramo vsoto kvadratov razdalj $\sum l_i^2 = \sum(dx_i^2 + dy_i^2) = \sum dx_i^2 + \sum dy_i^2$, lahko problem rešujemo neodvisno za x in y koordinato.

Če pogledamo pohištvo po naraščajočih številih, bodo v končni rešitvi določene strnjene skupine pohištva na isti koordinati, skupine pa bodo po koordinatah naraščale. Ugotoviti moramo, kako razdeliti pohištvo v skupine in kam postaviti neko skupino.

Drugi problem je lažji. Pohištvo na koordinatah $x_1, ..., x_m$ je optimalno premakniti na njihovo povprečno vrednost $\bar{x} = \frac{1}{m}\sum x$. V to se lahko prepričate z zapisom funkcije, ki predstavlja ceno premika na x, in jo minimizirate preko odvoda.

Prvi problem rešujemo postopoma z dodajanjem novega pohištva od 1 do n in prilagajanjem rešitve. Če se i-ti kos na mestu x_i nahaja desno od zadnje skupine, nimamo problema in ustvarimo novo skupino. Sicer ga dodamo zadnji skupini. Morda se je zaradi tega spremenila optimalna lokacija skupine, ki je sedaj bolj levo od predzadnje, zato ti dve skupini združimo itd. Potrebna podatka o neki skupini sta samo vsota koordinat in število elementov. Časovna zahtevnost rešitve je linearna.