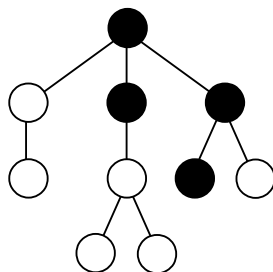


Algos

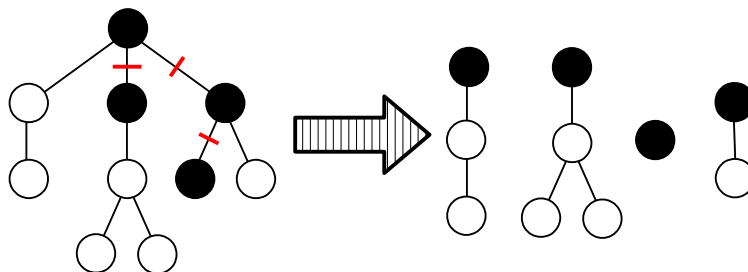
1 Svarstymai

Direktorius uždirba daugiau nei bet kuris kitas darbuotojas, todėl direktoriaus atlyginimas yra n . Todėl galime laikyti, kad direktoriaus alga yra vieša.

Žymėkime įmonės hierarchiją medžiu, kurio šaknis yra direktorius, ir visas viršūnės žymėkime apskritimais. Užspalvinkime tuos apskritimus, kurie žymi *viešus* darbuotojus (t.y. tuos, kurių algos yra viešos). Dėl patogumo vadinkime užspalvintus apskritimus *juodomis*, o neužspalvintus — *baltomis* viršūnėmis. Pavyzdžiui, galėtume turėti tokį grafą:

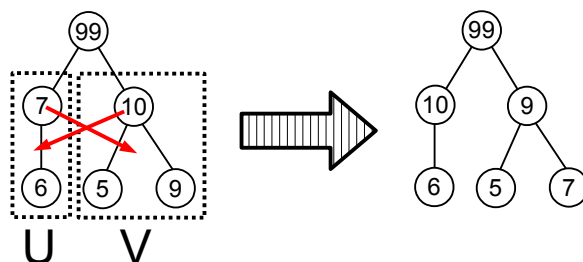


Kadangi pradiniai duomenys yra korektiški, mums nereikia atsiminti, kas yra viešo darbuotojo viršininkas. Taigi galimi nutrinti visas briaunas tarp juodų viršūnių ir gauti mišką, kurio kiekviena komponentė yra medis su juoda šaknimi ir likusiomis viršūnėmis baltomis:

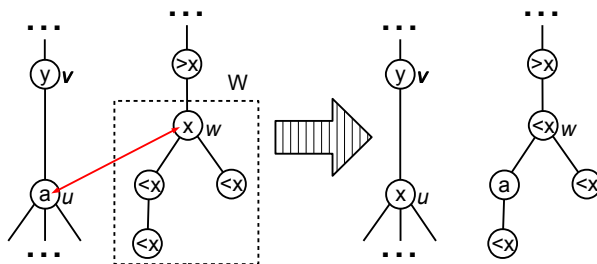


Sakysime, kad balta viršūnė turi *baltą seserį*, jei jos tėvinė viršūnė turi bent du baltus vaikus. Jei balta viršūnė neturi baltos sesers, sakysime, kad ji yra *balta vienturtė*.

Nesunku pajauti, kad mes galime nustatyti tik baltų vienturčių viršūnių algas. Iš tiesų, jei viršūnė u turi baltą seserį v , pažymėkime jų pomedžius U ir V , atitinkamai. Jei yra galimas algų paskirstymas, kai u gauna x , o v gauna y , tai perkeltume algą x į V , perkeltume algą y į U ir permaišykime algas U ir V , kad viršūnėms gautų daugiau už pavaldinius (taip padaryti visada galime), ir taip gausime naują galimą algų paskirstymą, kuriame u gaus kitokią algą nei x . Kitaip sakant, u algos vienareikšmiškai nustatyti negalime.



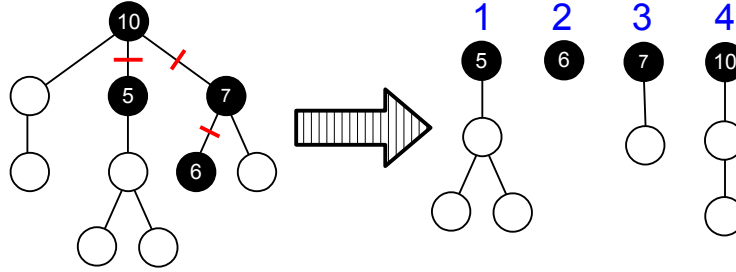
Taip pat nesunku pajauti, kad baltos vienturtės algą nustatyti galime, tik jei galime nustatyti jos tėvinės viršūnės algą. Iš tiesų, tarkime, kad baltos vienturtės u tėvinė viršūnė v gali įgyti skirtingas reikšmes x ir y , ir abejais atvejais u įgyja reikšmę a , kur $a < x < y$. Tegul w būna viršūnė, kuri įgyja reikšmę x , kai v įgyja reikšmę y . w negali būti toje pačioje miško komponentėje kaip u ir v . Tegul W žymi pomedį, kurio šaknis w (w yra balta viršūnė). Tuomet paskirstykime algas taip, kad u uždirbtų a , v uždirbtų y , ir w uždirbtų x . Sukeiskime u ir w algas ir pertvarkykime pomedį W , kad algos jame būtų išrikiuotos teisingai. Taip gausime galimą paskirstymą, kuriame u uždirba $x \neq a$.



Šitas samprotavimas taip pat parodo, kad jei baltos vienturtės alga yra viena-reikšmiškai a , o jos tėvinės — y , tai visi sveikieji skaičiai x griežtai tarp a ir y yra juodų viršūnių algos.

2 Sprendimas

Išrikiuokime medžius (t.y. miško komponentes) šaknies algos didėjimo tvarka. Šia tvarka medžius ir nagrinėsime.



Nagrinėdami medį t pasižiūrime, kiek viršūnių jis iš viso turi (tai galime būti apskaičiuavę iš anksto, bet nebūtina) ir apskaičiuojame, kiek viršūnių bendrai turi medžiai, kuriuos nagrinėjome (t.y. medžiai $1, 2, \dots, t$). Kadangi pradiniai duomenys yra korektiški, ši suma (pažymėkime ją S) niekada nebus didesnė už medžio t šaknies algą (pažymėkime ją A), taigi $S \leq A$. Taip pat žymėkimės, su koku didžiausiu $j < t$, medį j sudaro ne vien tik jo šaknis. Tegul b būna medžio j šaknies alga. Jei j nėra apibrėžtas, t.y. jei visi medžiai $1, 2, \dots, t - 1$ susideda vien tik iš šaknies, laikykime, kad $b = 0$.

Jei medžio t šaknis turi bent du vaikus, negalime nustatyti jų algos, todėl einame prie medžio $t + 1$. Laikykime, kad medžio t šaknis turi lygiai vieną vaiką. Jei $S < A$, tai tegul p ir q būna du didžiausi skaičiai, mažesni už A ir nesantys juodų viršūnių algos. Tuomet panaudoję ankstesnį argumentą matome, kad medžio t šaknies vaiko alga gali būti tiek p , tiek q . Tokiu atveju vėl nieko negalime pasakyti apie medžio t viršūnių algas ir einame prie medžio $t + 1$.

Galiausiai tarkime, kad $S = A$ ir medžio t šaknis turi tik vieną vaiką v_1 . Tokiu atveju galime vienareikšmiškai nustatyti, kad v_1 alga yra didžiausias skaičius, mažesnis už A ir nesantis juodos viršūnės alga. Tegul šis skaičius būna A_1 . Jei v_1 turi kitokių skaičių vaikų nei vienas, einame prie medžio $t + 1$. Jei v_1 turi lygiai vieną vaiką v_2 , tegul A_2 būna didžiausias skaičius, mažesnis už A_1 ir nesantis juodos viršūnės alga. Jei $A_2 > b$, priskiriame v_2 algą A_2 . Tęsiame tą patį su $v_2, v_3, \dots$, kol pasiekiamo viršūnę v_k , kuri arba turi ne lygiai vieną vaiką, arba $A_{k+1} < b$. Tuomet einame prie medžio $t + 1$.

Kai išnagrinėjame visus medžius, baigiame vykdymą. Nesunku įsitikinti, kad šio algoritmo sudėtingumas yra $O(n)$.