Drvosjeca

Kompleksnost O(n log n)

#grafovi #greedy #binarna\_pretraga #dfs

Prvo moramo izbjeci situaciju da premjestimo jednu granu i onda kasnije premjestamo grane koje se nalaze u njenom podstablu. To postizemo tako sto cemo pomjerati podstabla unazad, to jeste prvo one sa najvecom dubinom. Ako znamo da zelimo da dobijemo visinu stabla h, moramo se uvjeriti da sjecemo podstabla sa najvecom dubinom h – 1 (jer dobijamo +1 dubine kad ga premjestimo na korijen). Ako posjecemo podstablo dubine h, kada ga premjestimo na korijen imacemo podstablo dubine h + 1, i posto smo rekli da idemo unazad, vise se ne mozemo vratiti da opet posjecemo to stablo, tako da nismo popravili rezultat. Zbog toga izbjegavamo tu situaciju.

Ako fiksiramo visinu stabla h, mozemo se zapitati koliko treba operacija da se dobije stablo dubine h. Lako je vidjeti, ako se povecava h, da ce nam trebati manje operacija. To znaci da je rjesenje moguce naci **binarnom pretragom**.

Ako je visina stabla veca od h trenutno, znaci ima neki cvor koji ima dubinu vecu od h. Ako posmatramo to podstablo, najbolje bi bilo posjeci na mjestu gdje je on pa h – 1 cvorova na gore. Kada uzmemo i h – 1 cvor, onda ce dubina tog podstabla biti tacno h – 1. Znaci strategija je da sjecemo h – 1 roditelj najdubljeg cvora dok dubina stabla ne postane najvise h.

Da nadjemo h – 1 roditelja, mozemo lako izracunati unaprijed pomocu dfs prije svake iteracije odrzavajuci stek cvorova koje smo posjetili. Jos ostaje kako upamtiti da li je ivica uklonjena ranije. To radimo takodje sto pokrenemo dfs svaki put nakon odsijecanja i cuvamo u nizu posjeta. Vise u kodu.