

PATRIK

Zbog velikog ograničenja na N , kvadratno rješenje nije dovoljno efikasno.

Radi jednostavnosti, pretpostavimo za početak da su sve osobe različitih visina. Zamislimo da prolazimo kroz niz od prve osobe do zadnje. Promotrimo bilo koju osobu A u nizu. Ako u prolasku kroz niz nakon osobe A naiđemo na višu osobu B , tad osoba A sigurno ne vidi nikoga nakon osobe B .

Zbog toga u prolasku kroz niz možemo održavati stog "otvorenih potproblema", gdje je otvoreni potproblem osoba koju smo već prošli u nizu, a za koju još postoji mogućnost da može vidjeti neku osobu na koju još nismo naišli. Očito je da su potproblemi na stogu uvijek sortirani silazno po visini.

Kad naiđemo na novu osobu, uočimo da se ona vidi sa svim osobama na stogu od kojih je viša, te ujedno i skida te osobe sa stoga (jer zatvara njihove potprobleme). Ako stog ne ostane prazan, također se vidi s prvom višom osobom na stogu. Tada ulazi na stog (jer sigurno postoji mogućnost da vidi neku osobu kasnije u nizu).

Iako ćemo u rješenju imati dvije ugnježdene petlje, ukupna složenost je $O(N)$, jer svaka osoba točno jednom ulazi na stog i jednom silazi s njega, a svaka iteracija unutarnje petlje miče neku osobu sa stoga.

Za potpuno rješenje potrebno je još razmotriti utjecaj osoba jednakih visina. Jedan način je na stogu držati parove (visina, broj osoba) te ga takvoga održavati.

POLICIJA

Zadatak se rješava izgradnjom DFS stabla, te izračunavanjem standarnih vrijednosti za svaki čvor prilikom obilaska stabla, a to su:

- Discovery time – vrijeme prvog posjeta čvoru
- Finish time – vrijeme zadnjeg posjeta čvoru
- Depth – dubina čvora u stablu
- LowLink – Čvor s najmanjim discovery time-om do kojeg je moguće doći iz danog čvora preko jednog back edge-a.

Iz gornjih podataka moguće je izgraditi funkcije kojima možemo pitati je li čvor A u stablu predak čvoru B , te ako je i pronaći dijete čvora A koje je također predak čvoru B .

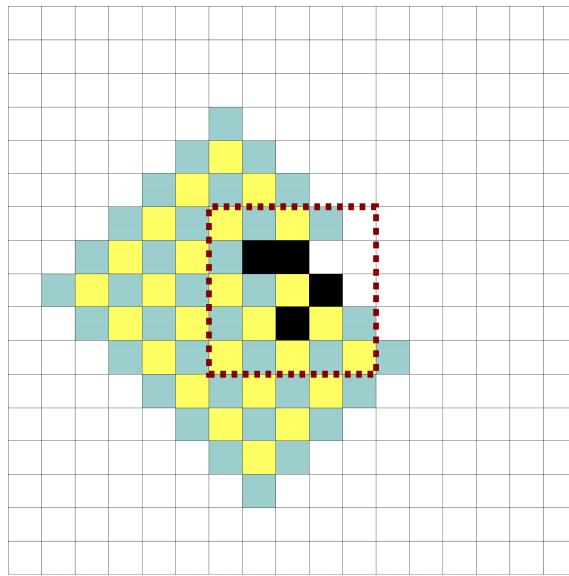
Te funkcije, zajedno sa prethodno izračunatim vrijednostima dovoljne su za odgovaranje na oba tipa pitanja. Za implementacijske detalje, vidi izvorne kodove.

SABOR

Prvi korak do rješenja je prepoznavanje da se u zadatku zapravo traži (među svim poljima udaljenim najviše K koraka) broj polja udaljenih paran broj koraka, te broj polja udaljenih neparan broj koraka od početnog polja. Možemo također reći da polja bojamo u dvije boje.

Boja polja je jednoznačno određena njegovim koordinatama jer svaki korak mijenja parnost izraza $x+y$. Zbog toga su dva susjedna polja nužno različite boje. Dodatno, udaljenosti dvaju susjednih polja od početnog se nužno razlikuju za točno 1.

Na slici je drugi primjer iz zadatka, s malo većim brojem K . Uočimo najmanji pravokutnik koji obuhvaća sve prepreke i početno polje, te ga proširimo jedno polje u svim smjerovima:



Zbog toga što su koordinate prepreka najviše 1000 po apsolutnoj vrijednosti, udaljenosti (a time i boje) svih polja unutar označenog pravokutnika možemo naći pretraživanjem u širinu.

Ukoliko je polje npr. na lijevom rubu tog pravokutnika (s unutarnje strane), udaljeno D koraka od početnog polja, tad je obojano još točno $K-D$ polja lijevo od njega izvan pravokutnika, te je moguće jednostavnim formulama odrediti broj polja jedne te broj polja druge boje. Slično vrijedi za polja na ostalim rubovima.

Potrebno je još prebrojati polja u kutevima, npr. polja gore i lijevo od žutog polja u gornjem lijevom rubu pravokutnika. Nazire se trokutasti uzorak te je moguće naći formule (vidi priložen kôd) koje daju broj polja po bojama.