



## Problema Karb – Soluție

*Andrei Ciocan, Student Universitatea Politehnică, București*

Problema se poate rezolva prin intermediul programării dinamice. Vom încerca să pornim de jos, începând cu subarbori cât mai mici și apoi să urcăm în arbore.

În încercarea colorării unui subarbor, observăm că apar 2 cazuri: rădăcina subarborului este influențată de un nod care se afla în subarbor, sau influențat de un nod care nu aparține subarborului (apare în sus). Astfel vom obține o dinamică:

- $din[nod][0]$  = numărul de noduri astfel încât nod este influențat de un cartel din subarbor

- $din[nod][1]$  = numărul de noduri astfel încât nod este influențat de un cartel care nu aparține subarborului

Recurența se poate obține în felul următor:

$$-din[nod][1] = \prod (din[fiu][0] + din[fiu][1])$$

- $din[nod][0] = \sum_i din[i][0] * \prod (din[fiu][0] + din[fiu][1])$ , unde  $i$  este fiu al lui  $nod$ ;  $fiu$  este de asemenea copil al lui  $nod$ , dar diferit de  $i$ . Ideea este că alegem ca nod să fie influențat de același cartel care influențează și fiul  $i$  (ceilalți copii ai nodului  $nod$  putând fi influențați în orice mod).

Se pot diferenția și cazuri speciale, atunci când un nod este cartel,  $din[nod][1] = 0$  (orice nod din subarbor nu poate fi influențat de un nod din afara subarborului, întrucât nu au acces). În acest caz, când nodul este cartel, subarborii fii se pot colora în orice fel, deci  $din[nod][0] = \prod (din[fiu][0] + din[fiu][1])$ .

În final, rezultatul se va afla în  $din[1][0]$ .

Pentru această abordare, se obțin în jur de 70 de puncte. Pentru a obține punctajul maxim, trebuie optimizată formula de calcul al numărului de configurații când nodul rădăcina este influențat de un nod din subarbor.

Una din soluții ar fi să precalculăm produsul tuturor fiilor unui nod,  $produsPrecalculat = \prod (din[fiu][0] + din[fiu][1])$ , iar la calcularea sumei înmulțim cu inversul modular al factorului respectiv lui  $i$ :  $(din[i][0] + din[i][1])$ , unde  $i$  este în formula de mai sus. Vom avea astfel:  $din[nod][0] = \sum produsPrecalculat * inversModular(din[i][0] + din[i][1])$ . Trebuie avut grijă când  $din[i][0] + din[i][1]$  este 0, întrucât nu putem împărți la el. Altă abordare ar fi prin programare dinamică:  $left[i] = \prod (din[fiu][0] + din[fiu][1])$ , unde  $fiu$  este un fiu de la stânga lui  $i$ , și  $right[i] = \prod (din[fiu][0] + din[fiu][1])$ , cu  $fiu$  la dreapta lui  $i$ .

Nu vom avea decât să adăugăm la sumă  $left[i] * din[i][0] * right[i]$ .