**Problema Karb – Soluţie**

*Andrei Ciocan, Student Universitatea Politehnică, Bucureşti*

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

În încercarea colorării unui subarbore, observăm că apar 2 cazuri: rădăcina subarborelui este influenţată de un nod care se afla în subarbore, sau influenţat de un nod care nu aparţine subarborelui (apare în sus). Astfel vom obţine o dinamică:

-din[nod][0] = numărul de noduri astfel încât nod este influenţat de un cartel din subarbore

-din[nod][1] = numărul de noduri astfel încât nod este influenţat de un cartel care nu aparţine subarborelui

Recurenţa se poate obţine în felul următor:

-din[nod][1] =

-din[nod][0] = , 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 sa fie influenţat de acelaşi cartel care influenţează şi fiul i (ceilalţi copii ai nodului nod putând fi influenţaţi în orice mod).

Se pot diferenţia şi cazuri speciale, atunci când un nod este cartel , din[nod][1] = 0 (orice nod din subarbore nu poate fi influenţat de un nod din afara subarborelui, întrucât nu au acces). În acest caz, când nodul este cartel, subarborii fii se pot colora în orice fel, deci din[nod][0] = .

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

Pentru această abordare, se obţin în jur de 70 de puncte. Pentru a obţine punctajul maxim, trebuie optimizată formula de calcul al numărului de configuraţii când nodul rădăcina este influenţat de un nod din subarbore.

Una din soluţii ar fi să precalculam produsul tuturor fiilor unui nod, produsPrecalculat =, iar la calcularea sumei înmulţim 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] = . Trebuie avut grijă când din[i][0] + din[i][1] este 0, întrucât nu putem împărţii la el. Altă abordare ar fi prin programare dinamică: left[i] = , unde fiu este un fiu de la stânga lui i, şi right[i] = , cu fiu la dreapta lui i.

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