Problema excursie - descrierea soluției

Alex Cociorva - Universitatea Babeș Bolyai, Cluj-Napoca
Adrian Panaete - Colegiul National A.T. Laurian, Botosani

Vom considera graful orientat format din elevii liceului. Pentru fiecare elev **`i`**, va exista o muchie orientata de la **`i`** la **`pref[i]`**.

Observatia principala este ca acest graf are o proprietate speciala: este un “graf soare”. Mai exact, va avea doar componente conexe cu urmatoarea forma:

- un ciclu de lungime >= 2.
- 0 sau mai multi arbori “agatati” de unele noduri ale ciclului.

O alta observatie este ca mereu va fi optim sa cuplam in camera 2 elevi care se prefera reciproc. Graful ramas acum va avea doar cicli de lungime > 2 si arbori “agatati” sau nu de unele noduri ale ciclilor.

Putem renunta acum la orientarea grafului si sa privim fiecare muchie ca fiind unidirectionala. Ce dorim sa aflam acum este cuplajul maxim pe acest graf.

Solutie O(N) programare dinamica - Alex Cociorva

Pentru inceput putem considera doar arborii. Pentru un nod x din arbore, calculam urmatoarea dinamica: tree_dp[x][0/1] = cuplajul maxim pentru nodurile din subarborele lui x daca il cuplam sau nu pe x cu unul din fiii lui. Recurenta se obtine usor luand in considerare daca am cuplat fiii sau nu.

In caz ca arborii nu mai sunt agatati de niciun ciclu (au fost agatati de un ciclu de lungime 2 pe care l-am eliminat la inceput), atunci adunam solutia pentru acesti arbori la solutia finala.

Tot ce ne mai ramane de facut este sa folosim muchiile de pe ciclu. Astfel, luam fiecare ciclu si calculam urmatoarea dinamica:
cycle_dp[x][0/1] = cuplajul maxim daca il cuplam pe x sau nu, unde x e un nod de pe ciclu.

Avem 3 variante:

- il cuplam pe x cu elementul precedent de pe ciclu: cycle_dp[x][1] = 1 + cycle_dp[last][0] + tree_dp[x][0]
- il cuplam pe x cu un nod din arbore (in caz ca in x este agatat un arbore): cycle_dp[x][1] = tree_dp[x][1] + max(cycle_dp[last][1], cycle_dp[last][0])
- nu il cuplam pe x cu nimeni: cycle_dp[x][0] = tree_dp[x][0] + max(cycle_dp[last][1], cycle_dp[last][0])

Trebuie avut grija la dinamica intr-un cat trebuie considerat tot ciclul. O varianta ar fi sa alegem o muchie de pe ciclu si sa o consideram cuplata sau nu. Ce ramane este un lant pe care poate fi aplicata dinamica de mai sus.

Solutie O(N) greedy - Adrian Panaete

Precalculam la citire dorit[i] = numarul de copii cu pref[k]=i (cati ar dori sa stea in camera cu i).
Folosim un vector cuplaj[i] = elevul cu care sta in camera elevul i.

Se face repartizarea in doua faze:

<u>FAZA 1</u>

Cream o coada cu elevi necuplati care au dorit[i] = 0;
Procesam elevii din coada.
Pentru un elev i procesat fie j=pref[i] si k=pref[j]. Sunt 3 cazuri:

- daca i este deja cuplat nu facem nimic.
- daca i nu e cuplat dar j e cuplat, i va ramane oricum nefericit, deci nu facem nimic.
- daca i si j nu sunt cuplati, ii cuplam si prin asta trebuie sa decrementam dorit[k]. Observam ca daca dupa decrementare k nu mai e dorit de nimeni si e necuplat atunci trebuie sa il introducem in coada.

<u>FAZA 2</u>

Elevii ramasi sunt fie cei ramasi necuplati in faza 1 (si acestia oricum vor ramane nefericiti), fie elevi care formeaza cicli in care oricare elev isi prefera succesorul din ciclu.
Formam cuplaje de cate doi consecutivi din ciclu obtinand un copil fericit si unul nefericit.
La final daca ciclul are lungime impara, cel necuplat din ciclu va ramane pana la final nefericit.

Pentru a determina fericirea maxima parcurgem simultan vectorii pref si cuplaj si incrementam fericirea pentru fiecare elev i pentru care pref[i]=cuplaj[i].