

Tj – descrierea soluției

Se spune ca un nod A domina un nod B daca exista muchia A-B si oricare ar fi un alt nod C, astfel incat C este un vecin al lui B, C este si vecin al lui A (nodul A poate avea si alti vecini, in afara de B si vecinii lui B). Un nod Y se numeste *dominat* daca exista cel putin un nod X, astfel incat X domina pe Y.

Se constata usor ca, in cazul in care Tom are strategie sigura de castig, la sfarsitul jocului, inaintea ultimei mutari a lui Jerry, Tom se afla intr-un nod X si Jerry se afla intr-un nod Y, iar nodul Y este dominat de nodul X (aceasta este echivalent cu a spune ca orice mutare ar efectua Jerry, el va fi prins de Tom la urmatoarea mutare a lui Tom).

Este evident ca daca graful dat nu contine nici un nod dominat (si are mai mult de un singur nod), Tom nu are strategie sigura de castig (deoarece Jerry va putea efectua de fiecare data o mutare care sa il “scape”).

Observatia care sta la baza rezolvarii problemei este ca, in cazul in care exista un nod dominat in graf, atunci Tom are strategie sigura de castig *daca si numai daca* el are strategie sigura de castig in graful din care se elimina nodul dominat (impreuna cu toate muchiile adiacente). Acest lucru se poate demonstra observand ca o strategie de castig pentru graful *redus* (din care s-a eliminat nodul dominat) se poate extinde la graful care contine nodul dominat (unele mutari de “stat” in nodul X din strategia pentru graful redus sunt “echivalente” cu unele mutari de deplasare din nodul X in nodul Y, unde nodul Y este nodul eliminat din graf si nodul X este unul din nodurile care il domina pe Y).

Folosind aceasta observatie, se poate folosi urmatorul algoritm :

- Cat timp graful are cel putin 2 noduri si contine cel putin un nod dominat:
gaseste un nod dominat si elimina-l din graf
- Daca graful a ramas cu un singur nod, atunci raspunsul este DA, altfel raspunsul este NU

Evident, acest algoritm poate fi implementat folosind mai multe abordari:

- algoritmul “naiv” are complexitatea $O(N^4)$, insa, in practica, are un timp de executie destul de bun, deoarece, in grafuri “dense” se gasesc usor noduri dominate, iar in grafuri cu putine muchii, algoritmul se poate implementa folosind liste de vecini, complexitatea acestuia depinzand acum de numarul de muchii
- solutia “oficiala” foloseste un algoritm cu complexitatea $O(N^3)$
- exista algoritmi cu complexitate mai buna (cautati “dismantlable graphs” pe Google :)), insa nu am gasit decat mentionari ale existentei acestora (nu si descrierile lor)