

Tema 2 Analiza Algoritmilor 2021

Responsabil: David Iancu

Data publicare: 03.12.2021

Deadline: 17.12.2021

1 Șah

Toată lumea știe cum este jocul de șah - ai o tablă de 8×8 , niște piese, iar scopul final este să capturezi regele adversarului. Pentru această problemă vom folosi o tablă de dimensiune $N \times N$ iar task-ul vostru este unul mult mai simplu - să plasați N regine pe tablă fără ca acestea să se atace. Pentru asta, trebuie să scrieți atât un program în limbajul C/C++, cât și un algoritm nedeterminist. Pentru ambele variante trebuie estimată complexitatea temporală. Pentru programul în limbajul C, se citește un singur număr, N , reprezentând dimensiunea tablei, și trebuie afișată o posibilitate de a aranja reginele (oricare) în felul următor: la ieșire se vor scrie N linii, reprezentând perechi de tipul (linie, coloană) pentru fiecare regină în parte. Pentru simplitate, putem considera că se citește din fișierul "sah.in" și se scrie în fișierul "sah.out".

2 Grafuri

Avem următoarea problemă - se dau K grafuri neorientate ca date de intrare și un număr $K2$. Există un subgraf de dimensiune $K2$ în primul graf care este izomorf cu un subgraf de dimensiune $K2$ din al doilea graf, cu un subgraf de dimensiune $K2$ din al treilea graf, și așa mai departe? Practic, ne dorim să găsim câte un subgraf de dimensiune $K2$ în fiecare dintre cele K grafuri, cu proprietatea că toate sunt izomorfe între ele. Ca și mai sus, faceți atât varianta deterministă, în limbaj C/C++, cât și varianta cu algoritmi nedeterminist și estimați complexitatea temporală pentru cei doi algoritmi. Pentru varianta deterministă, puteți considera că pe prima linie a fișierului "graf.in" se citește numărul K . În continuare, pentru fiecare din cele K grafuri se citesc câte 2 numere, V_i și E_i , reprezentând numărul de noduri și numărul de muchii al fiecărui graf. Pe următoarele E_i linii se citesc perechi de tipul (a,b), reprezentând faptul ca avem o muchie de la nodul a la nodul b. În fișierul "graf.out" se va afișa "-1" dacă nu există soluție, iar dacă există soluție se afișează pe câte o linie separată $K2$ numere, reprezentând indicii nodurilor din cele K grafuri care formează subgrafuri izomorfe.

Problema este NPC, astfel încât pentru varianta deterministă considerăm $K \leq 4$, $V_i \leq 8$, $K2 \leq 4$.

3 Reducere

Se dau următoarele 2 probleme.

Problema 1: Avem un oraș care are N case și M drumuri bidirecționale care conectează aceste case. Din păcate, drumurile s-au stricat și trebuie să fie reparate. Pentru a putea repara un drum, trebuie să existe materiale în cel puțin una dintre cele 2 case care conectează drumul respectiv. Pentru un

număr K dat, putem alege K case astfel încât să putem repara toate drumurile?

Problema 2: David are o colecție de N bacnote vechi, toate diferite între ele. Se știe că David e prieten cu mulți studenți. De curând a aflat numărul lor, M . Fiecare din cei M studenți dețin câte un subset din cele N bacnote pe care le are David. Acesta le dă un număr K și le spune că dacă vor 10 la temă, trebuie să se găsească doar K studenți din cei M care să dețină, împreună, toate tipurile de bacnote pe care le are David. Se pot găsi K astfel de studenți, știind tipurile de bacnote pe care le are David și tipurile de bacnote pe care le deține fiecare dintre cei M studenți? Cerință: Cele 2 probleme sunt NP. Pentru fiecare din cele 2 probleme se cere câte un algoritm nedeterminist. La final, demonstrați ca problema 1 se reduce polinomial la problema 2. De asemenea, să se estimeze complexitatea temporală pentru cei doi algoritmi.

4 Precizări suplimentare

- Temele nu se testează automat, astfel încât puteți folosi și alt format pentru datele de intrare/de ieșire dacă vă este mai comod. Formatul din cerință este dat ca exemplu.
- Pentru programele în limbaj C/C++, se așteaptă un program complet. Pentru problemele nedeterminate, se recomandă să le scrieți într-un document .txt. Documentul principal poate fi în format word/ libreOffice/ latex, pentru reducerea polinomială și pentru explicații suplimentare legate de fiecare problemă. Se acceptă teme scrise de mână doar în cazuri justificate (dacă aveți nevoie de elemente mai greu de scris într-un document word, cum ar fi grafice, desene, etc).
- Fiecare problemă are câte 3 puncte și există un punct din oficiu. În cadrul primelor 2 probleme, algoritmul nedeterminist valorează 1.5 puncte iar programul determinist (care trebuie să fie funcțional) valorează, de asemenea, 1.5 puncte. Pentru ultima problemă reducerea valorează 2 puncte iar algoritmi nedeterminiști câte 0.5 puncte. Pentru fiecare problemă estimarea complexității reprezintă 10% din punctaj.
- Pentru estimarea complexității este suficient să oferiți o limită superioară.
- În cazul în care vă este mai ușor, se acceptă să scrieți pseudocodul de mână pentru algoritmi nedeterminiști, dar nu este recomandat (nu se scade dar este mai ușor de corectat un algoritm scris în format txt).
- Deadline-ul este hard. Temele se vor încărca atât pe platforma Microsoft Teams, cât și pe Moodle.