

Analiza algoritmilor

Tema 2

Deadline: 18.12.2020

Ora 23:55

Responsabili temă: Mihai Nan, David Iancu

Profesor titular: Andrei-Horia Mogoș

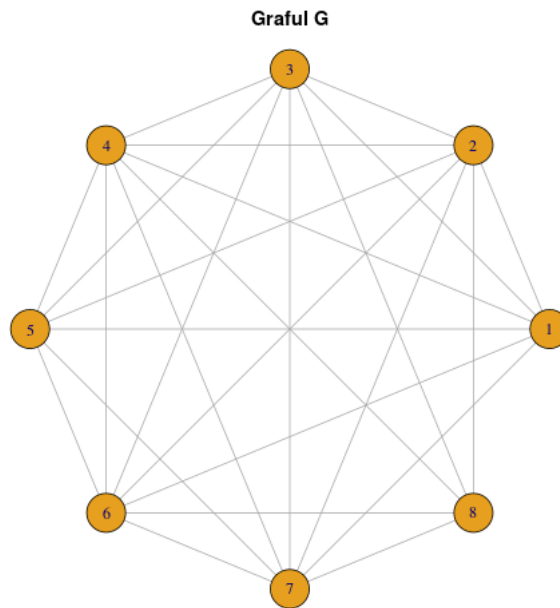
Facultatea de Automatică și Calculatoare
Universitatea Politehnica din București
Anul universitar 2020 - 2021
Seria CC

1 Problema 1 – Noțiuni teoretice

1. Fie următoarele definiții:

- O clică în graful neorientat $G = (V, E)$ este un subgraf complet al lui G . Adică, este un graf $G_1 = (V_1, E_1)$, unde $V_1 \subseteq V$ este o mulțime de noduri astfel încât fiecare două noduri din V_1 sunt cele două capete ale unui arc din G .
- O clică maximală este o clică la care nu se mai pot adăuga noduri. Pentru fiecare nod v , care nu face parte dintr-o clică maximală, trebuie să existe un alt nod w , care este în clică și nu este adiacent lui v , ceea ce împiedică adăugarea lui v la clică.
- O clică transversală maximală a unui graf neorientat $G = (V, E)$ este o submulțime de noduri S cu proprietatea că fiecare clică maximală a lui G conține cel puțin un vârf care se află în S .

Se oferă următorul graf G :



Cerințe:

- Verificați dacă există o 4-clică în graful oferit și dacă identificați una atunci desenați respectiva clică.
- Identificați clicile maxime care există în graful G .
- Oferiți un exemplu de clică transversală maximală a grafului G .

2. Se oferă următoarea formulă propozițională F în forma normală conjunctivă (CNF):

$$(A \vee B \vee C) \wedge (A \vee B \vee \neg C \vee D) \wedge (A \vee \neg B \vee C) \wedge (\neg A \vee B \vee \neg C) \wedge (A \vee B \vee \neg D) \wedge (A \vee \neg B \vee \neg D) \wedge (\neg A \vee B \vee D) \wedge (A \vee C \vee D \vee B)$$

Verificați dacă formula dată este sau nu satisfiabilă (este suficient să oferiți o asociere a variabilelor formulei F la valori din $0, 1$ astfel încât valoarea lui F să fie 1 sau să explicați de ce nu este satisfiabilă).

Determinați numărul de noduri din graful G_F echivalent formulei F (G_F se obține folosind algoritmul prezentat la curs). Care sunt nodurile din G_F cu care o să fie adiacent nodul $\langle A, 5 \rangle$?

2 Problema 2 – Algoritm nedeterminist polinomial

Toata lumea s-a jucat FreeCell când era mic – jocul presupune un pachet standard de cărți, împărțite aleator în mai multe coloane (8 coloane). Scopul jocului este să așezi cărțile în ordine de la cea mai mică valoare ($AS = 1$) până la cea mai mare ($K = 14$) în 4 stive puse separat (**Cleared Cards**). Pe lângă cele 8 coloane (**Columns**) și cele 4 stive, mai există 4 căsuțe (**Freecells**) în care se pot pune cărți, dacă nu există mutări disponibile, sau pentru a elibera o carte care nu e disponibilă.

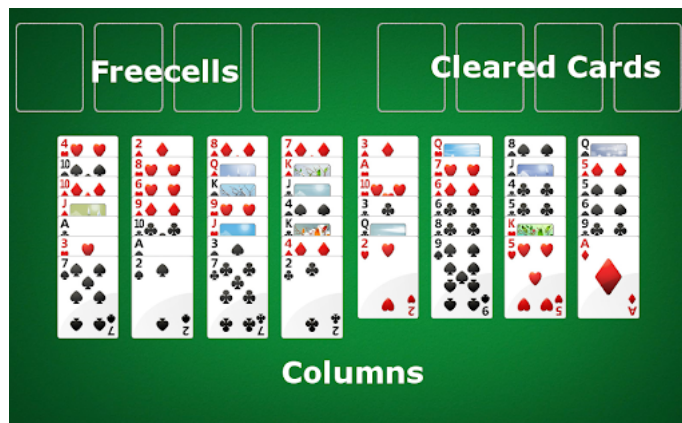


Figure 1: Configurația tablei de joc pentru acest joc

Există următoarele tipuri de mutări:

- O carte de culoare neagră se poate pune doar peste o carte de culoare roșie, de valoare imediat superioară și invers;
- Un grup de cărți (de exemplu 2, 3, 4, 5, 6) se poate pune peste o carte imediat superioară, de culoare opusă celei mai mari din grup. Lungimea grupului nu poate fi mai mare decât $(nr\ de\ căsuțe\ speciale\ libere + 1) * 2^{numar\ de\ coloane\ libere}$ (dacă se muta pe o coloană liberă, nu se consideră și coloana în cauză ca fiind liberă).
- O carte se poate pune în zona de căsuțe speciale, dar cu prețul de a nu mai putea muta grupuri la fel de mari de cărți.

Pentru o configurație dată, ne întrebăm dacă există soluție. Această problemă este în **NP** dacă vom extinde jocul de FreeCell la un pachet de N cărți, cu C coloane și M tipuri de căsuțe speciale. De asemenea, vom avea $M/2$ tipuri de cărți roșii și $M/2$ tipuri de cărți negre (nu doar inima neagră, inimă roșie, treflă, romb, dar în așa fel încât să se păstreze ideea jocului, să poți pune negru doar peste roșu și invers, dar să existe $M/2$ tipuri de cărți negre, $M/2$ tipuri de cărți roșii).

Recapitulând, avem M culori, din care $M/2$ roșii, $M/2$ negre, N cărți din fiecare culoare, M stive în care trebuie așezate cărțile, fiecare culoare separat, de la valoarea 1 la N , C coloane și M căsuțe speciale în care putem depozita cărți temporar.

O carte se poate defini ca:

```
struct Carte{
    int valoare;
    int culoare; //0 = rosu, 1 = negru de exemplu
    int tip_culoare; // de la 1 la m/2
};
```

O configurație se poate defini așa:

```
struct game {
    int C, N, M;
    Carte coloane[C][N];
    Carte freeCells[M];
    Carte homeCells[M][N];
}
```

Dându-se structura unui astfel de joc, să se determine dacă are soluție folosind un algoritm nedeterminist cu complexitate temporală polinomială (va trebuie să estimați complexitatea pentru algoritmul propus).

Important

Se poate ajunge în ciclu infinit, de aceea trebuie să rețineți și mulțimea stărilor în care a mai fost jocul până la momentul de timp t , pentru a nu le repeta.

3 Problema 3 – Reduceri polinomiale

3.1 Cerința 1

Fie $G = (V, E)$ un graf neorientat cu un număr nenul par de noduri și V' o submulțime de noduri cu proprietățile: $V' \subset V$ și V' conține exact jumătate din nodurile grafului G . O să numim **bisecția lui G în raport cu submulțimea V'** mulțimea $B_G(V') = \{(u, v) \in E \mid u \in V' \text{ și } v \in V \setminus V'\}$.

Pornind de la aceste informații, se definesc următoarele 2 probleme de decizie:

Problema Q_1 : Fie $G = (V, E)$ și un număr natural k cu proprietatea: $0 \leq k \leq \frac{|V|^2}{4}$. Există pentru graful G o submulțime de noduri V' , care conține jumătate din nodurile grafului G , astfel încât bisecția lui G în raport cu submulțimea V' conține cel mult k elemente?

Problema Q_2 : Fie $G = (V, E)$ și un număr natural k cu proprietatea: $0 \leq k \leq \frac{|V|^2}{4}$. Există pentru graful G o submulțime de noduri V' , care conține jumătate din nodurile grafului G , astfel încât bisecția lui G în raport cu submulțimea V' conține cel puțin k elemente?

Demonstrați că $Q_2 \leq_p Q_1$.

3.2 Cerința 2

Fie următoarea problemă numerică:

Problema Q_1 : Fie o matrice binară M de dimensiuni $m \times n$. Să se găsească un vector binar V de dimensiune n astfel încât $MV = \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix}$ (vector ce conține doar valori de 1).

Fie următoarea variantă simplificată a problemei TSP (Travelling salesman problem):

Problema Q_2 : Fie un graf neorientat $G = (V, E)$. Există un ciclu în graful G care trece prin toate nodurile o singură dată? (nu ne interesează și costul)

Demonstrați că problema Q_1 este reductibilă în timp polinomial problema Q_2 .

4 Punctaj

⚠ IMPORTANT !



Punctajul complet pe fiecare task se acordă doar dacă task-ul a fost **complet și corect** rezolvat. Tema valorează **1.5 puncte** din nota finală de la AA.

Problema	Punctaj
Problema 1.1	15 puncte
Problema 1.2	15 puncte
Problema 2	50 de puncte
Problema 3	70 de puncte

Atenție!

Tema este individuală! Toate soluțiile trimise vor fi verificate, folosind o unealtă pentru detectarea plagiatului. Orice parte a temei copiată, de pe Internet sau de la colegi, duce la anularea punctajului pentru temă. În cazul copierii de la un alt coleg se anulează și punctajul colegului respectiv pentru această temă.

Dacă o temă este uploadată după termenul limită, tema se anulează.

Observație

Se va acorda și punctaj parțial pe exercițiile propuse, în funcție de cât de mult s-a rezolvat corect.

5 Precizări generale pentru trimiterea temei

5.1 Cum se trimite tema?

Creați o arhivă cu denumirea **nume_prenume_grupa.zip**. De exemplu, studentul *Ionel Popescu* de la grupa *326CC* va crea arhiva *popescu_ionel_326CC.zip*.

Încărcați arhiva pe <https://curs.upb.ro/> până la data indicată (**18.12.2020, ora 23:55**). **NU** se acceptă teme trimise pe e-mail sau altfel decât prin intermediul platformei de curs.

Atenție! Formatul arhivei trebuie să fie **zip**.

5.2 Ce trebuie să conțină arhiva?

Un fișier **README**, având formatul **PDF**, care să fie semnat cu **nume**, **prenume** și **grupă**. Acest fișier trebuie să conțină răspunsul la întrebările și cerințele din enunț. Pentru redactarea fișierului **PDF**, puteți utiliza, de exemplu, \LaTeX sau orice procesor de documente pentru desktop sau online în care să puteți edita ecuații (exemple ar fi: Google Docs, LibreOffice Writer, Microsoft Word etc.) sau puteți rezolva exercițiile pe foi și să le scanați și să le adăugați într-un fișier **PDF**.

⚠ IMPORTANT !



Fișierul **README.pdf** trebuie să se afle în rădăcina arhivei și **NU** în alte foldere!

Atenție!

Orice întrebare legată de temă se va adresa pe forumul dedicat temei.