

# Analiza algoritmilor

Temă

**Deadline - Prima parte: 29.11.2017**

**Ora 23:55**

**Deadline - Partea a doua: 15.12.2017**

**Ora 23:55**

Responsabil temă: *Mihai Nan*

Profesor titular: *Andrei-Horia Mogoș*

Facultatea de Automatică și Calculatoare

Universitatea Politehnica din București

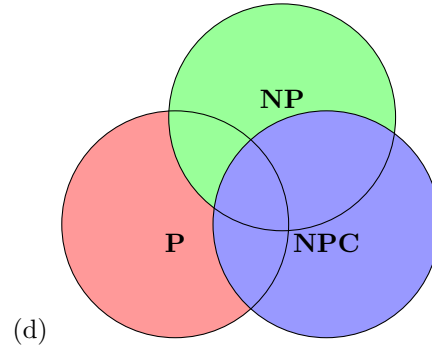
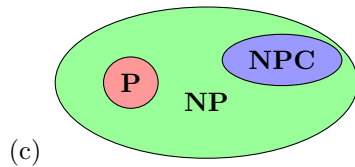
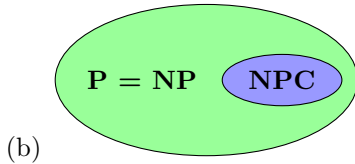
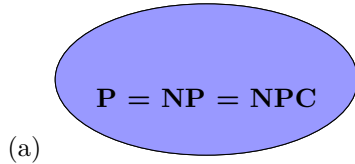
Anul universitar 2017 - 2018

Seria CC

# 1 Prima parte

## 1.1 Problema 1 - 10 puncte

Dându-se următoarele diagrame, explicați pentru fiecare dacă infirmă sau nu informațiile prezentate la curs pentru clasele de probleme **P**, **NP** și **NPC**.



## 1.2 Problema 2 - 40 de puncte

Roboțelul Wall-e este blocat într-un labirint 3D din care dorește să evadeze. Acest labirint are forma unui paralelipiped dreptunghic și conține mai multe tipuri de celule: celule goale (**G**), prin care robotul poate trece, celule blocate (**B**), prin care robotul nu poate trece, ieșiri (**I**), pe care robotul le poate folosi pentru a evada din labirint, celule care conțin recompense (fiecare astfel de celulă va conține un număr natural care semnifică valoarea recompensei), pe care robotul le poate colecta, și celule capcană (**C**), din care robotul nu mai poate evada, rămânând blocat în acestea pentru totdeauna. Cunoșcând coordonatele poziției în care se află robotul inițial,  $(x, y, z)$ , și știind că acesta se poate deplasa în față, în spate, stânga, dreapta, sus sau jos cu câte o celulă la fiecare pas, trebuie să îl ajutați să determine calea pe care trebuie să o urmeze pentru a părăsi labirintul colectând recompensa maximă posibilă.

(a) Propuneți un algoritm determinist care să rezolve această problemă. Calculați detaliat complexitatea temporală pentru algoritmul propus.

(b) Propuneți un algoritm nedeterminist care să rezolve această problemă. Calculați detaliat complexitatea temporală pentru algoritmul propus.

## 1.3 Problema 3 - 30 de puncte

Fie următoarele probleme de decizie:

**Short Path:** Fie  $G = (V, E)$  un graf neorientat ponderat, având costuri pozitive,  $u$  și  $v$  două noduri din graf și  $k$  un număr natural. Există în graf un drum de la  $u$  la  $v$  având costul mai mic sau egal cu  $k$ ?

**Long Path:** Fie  $G = (V, E)$  un graf neorientat ponderat, având costuri pozitive,  $u$  și  $v$  două noduri din graf și  $k$  un număr natural. Există în graf un drum de la  $u$  la  $v$  având costul mai mare sau egal cu  $k$ ?

(a) Să se stabilească dacă **Short Path**  $\in$  **P**. Să se justifice (demonstreze) răspunsul.

(b) Să se stabilească dacă **Long Path**  $\in$  **NP**. Să se justifice (demonstreze) răspunsul.

## 2 Partea a doua

### 2.1 Problema 4 - 20 de puncte

Găsiți o problemă din **NPD** care se poate reduce polinomial la **Long Path** și arătați cum se poate realiza această reducere.

O posibilă problemă pe care o puteți folosi în cadrul rezolvării este următoarea.



Dându-se un graf  $G = (V, E)$  să se determine dacă există un drum  $P$  în  $G$  care să treacă prin fiecare vârf exact o singură dată.

### 2.2 Problema 5 - 40 de puncte

Sherlock Holmes are de rezolvat un caz foarte important. Acesta a aflat că profesoul James Moriarty este liderul unui clan al criminalității organizate din care fac parte mai mulți criminali. După mai multe zile de investigație, a reușit să alcătuiască un cerc de  $n$  suspecți, iar pentru fiecare doi indivizi din acest cerc el a determinat care este probabilitatea ca amândoi să nu fie aliați în clanul lui Moriarty. Sherlock a apelat la unul din prietenii săi matematicieni pentru a afla care este probabilitatea totală maximă admisă  $K$  astfel încât să fie sigur că a determinat corect componența clanului criminal. Pentru a afla cu certitudine care este componența clanului criminal, Sherlock Holmes trebuie să determine indivizii pe care îi poate alege astfel încât suma probabilităților să fie mai mică sau egală decât  $K$ , iar numărul indivizilor aleși să fie și el maxim. Spre exemplu, dacă vom considera că din clan fac parte  $C_1, C_2$  și  $C_3$ , atunci probabilitatea totală o să fie  $Prob(C_1, C_2) + Prob(C_1, C_3) + Prob(C_2, C_3)$ .

Arătați că dacă Sherlock Holmes reușește să alcătuiască un algoritm determinist cu complexitate temporală polinomială care să găsească membrii clanului criminal condus de Moriarty atunci el rezolvă și o altă problemă importantă a informaticii teoretice, demonstrând că **P = NP**.

Puteți folosi în demonstrație următoarea problemă care este **NPC**.



Fiind date un graf neorientat  $G$  și un  $k$ , să se determine dacă există un subgraf format dintr-o submulțime a nodurilor grafului inițial, având  $k$  elemente, cu proprietatea că vârfurile nu sunt două câte două adiacente. Altfel spus, graful indus de aceste vârfuri are doar vârfuri izolate.

### 2.3 Problema 6 - 20 de puncte

Fie **Problema damelor**: Se dau  $N$  dame și o tablă de șah de dimensiune  $N \times N$ . Să se găsească toate modalitățile de a aranja toate damele astfel încât oricare două dame să nu se atace. Două dame se atacă dacă se află pe aceeași linie, coloană sau diagonală. Se cere să se afișeze toate soluțiile problemei.

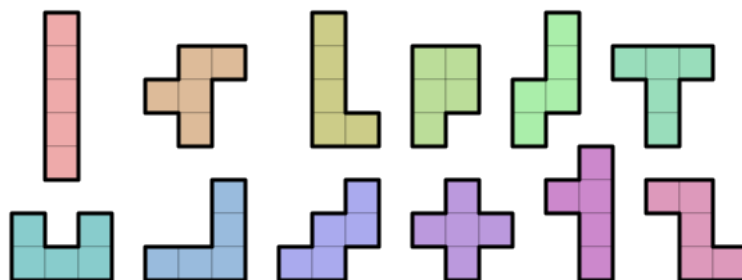
(a) Propuneți un algoritm determinist care rezolvă această problemă. Calculați detaliat complexitatea spațială pentru algoritmul propus.

(b) Propuneți un algoritm nedeterminist care rezolvă această problemă. Calculați detaliat complexitatea spațială pentru algoritmul propus.

### 2.4 Problema 7 - 40 de puncte

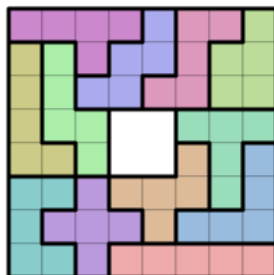
Un pentomino (sau 5-omino) este un poliomino de ordinul 5, adică un poligon în plan format din 5 pătrate de dimensiuni egale conectate. Atunci când se consideră că aplicând rotații și reflexii nu se obțin forme distincte, există 12 pentomino-uri unice posibile.

Fie problema **5-Omino Tiling Puzzle**: Dându-se cele 12 pentomino-uri distincte și o configurația sub forma unei matrice  $n \times m$  care conține valori de 0 și 1, determinați dacă există o combinație posibilă care să acopere toate pozițiile care conțin valoarea 1 și să le lase libere pe cele care conțin valoarea 0, utilizând cele 12 figuri puse la dispoziție.



**Explicații:** Cele 12 pentomino-uri distincte sunt următoarele:

Iar o soluție posibilă a problemei pentru  $n = m = 8$  este următoarea:



Configurația oferită este următoarea:

```

1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1
1 1 1 0 0 1 1 1
1 1 1 0 0 1 1 1
1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1

```

- Demonstrați că problema **5-Omino Tiling Puzzle** este în **NP**.
- Arătați că problema **5-Omino Tiling Puzzle** se poate reduce polinomial la problema **Set cover**.

#### Observație

Problema **Set cover**: Dându-se un număr natural  $k$ , o mulțime finită  $U = \{u_1, u_2, \dots, u_n\}$  și  $\mathcal{F}$  o familie de  $m$  submulțimi ale lui  $U$  a căror reuniune este egală cu  $U$ , determinați dacă există o subfamilie  $\mathcal{S} \subset \mathcal{F}$  formată din  $k$  submulțimi a căror reuniune să fie egală cu  $U$ .

### 3 Punctaj

⚠ IMPORTANT !



Punctajul complet pe fiecare task se acordă doar dacă task-ul a fost **complet** și **corect** rezolvat.  
Tema valorează **2 puncte** din nota finală de la AA.

Problema	Punctajul
Problema 1	10 puncte
Problema 2	40 de puncte
Problema 3	30 de puncte
Problema 4	20 de puncte
Problema 5	40 de puncte
Problema 6	20 de puncte
Problema 7	40 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.

Dacă veți încărca în arhiva pentru a doua parte a temei rezolvări ale problemelor din prima parte, acestea nu se vor lua în considerare.

### Observație

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

## 4 Precizări generale pentru trimiterea temei

### 4.1 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, cu soluțiile problemelor din prima parte, pe [cs.curs.pub.ro](http://cs.curs.pub.ro) până la data indicată (**29.11.2017, ora 23:55**). **NU** se acceptă teme trimise pe e-mail sau altfel decât prin intermediul platformei de curs.

Încărcați arhiva, cu soluțiile problemelor din a doua parte, pe [cs.curs.pub.ro](http://cs.curs.pub.ro) până la data indicată (**15.12.2017, 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**.

### 4.2 Ce trebuie să conțină arhiva?

Un fișier **README.pdf**, 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, **L<sup>A</sup>T<sub>E</sub>X** 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.).

### ⚠ IMPORTANT !

⚠ Fișierul **README.pdf** trebuie să se afle în rădăcina arhivei și **NU** în alte foldere!  
 Fișierul **README.pdf** **NU** poate conține imagini cu rezolvările exercițiilor realizate pe foi.