ALGORITMICA GRAFURILOR
Săptămâna 1

C. Croitoru

croitoru@info.uaic.ro

FII

October 1, 2013

1. **Descrierea cursului**
2. **Interesul penru grafuri în Informatică**
3. **Elemente introductive de complexitate**
4. **Problemele pentru seminarul 1**

Pagina cursului

http://thor.info.uaic.ro/~croitoru/ag/

Pagina cursului

http://thor.info.uaic.ro/~croitoru/ag/

Obiective

Studenții vor fi familiarizați cu noțiunile și rezultatele de bază ale **Teoriei Algoritmice a Grafurilor**, care vor fi aplicate în proiectarea de algoritmi eficienți pentru diverse probleme de optimizare combinatorică.

DESCRIEREA CURSULUI

Pagina cursului

http://thor.info.uaic.ro/~croitoru/ag/

Obiective

Studenții vor fi familiarizați cu noțiunile și rezultatele de bază ale **Teoriei Algoritmice a Grafurilor**, care vor fi aplicate în proiectarea de algoritmi eficienți pentru diverse probleme de optimizare combinatorică.

Tematică Generală

Clase de Complexitate, Vocabular al Teoriei Grafurilor, Probleme de drum(parcurgeri, drumuri minime, conexiune), Arbori parțiali de cost minim (union-find, complexitate amortizată), Cuplaje, Fluxuri, Reduceri polinomiale pentru probleme de decizie pe grafuri, Abordări ale problemelor NP-dificile, Grafuri Planare.

Competențe acumulate

Utilizarea grafurilor ca limbaj de modelare formală. Cunoașterea algoritmilor de bază pentru problemele clasice pe grafuri. Recunoașterea complexității de calcul pentru probleme de optimizare.

Competențe acumulate

Utilizarea grafurilor ca limbaj de modelare formală. Cunoașterea algoritmilor de bază pentru problemele clasice pe grafuri. Recunoașterea complexității de calcul pentru probleme de optimizare.

Metode de predare

Prezentari video ale slide-urilor (conținând notele de curs) disponibile in format pdf la inceputul semestrului.
http://thor.info.uaic.ro/~croitoru/ag/ag 13-14 allinone.pdf

Competențe acumulate

Utilizarea grafurilor ca limbaj de modelare formală. Cunoașterea algoritmilor de bază pentru problemele clasice pe grafuri. Recunoașterea complexității de calcul pentru probleme de optimizare.

Metode de predare

Prezentari video ale slide-urilor (conținând notele de curs) disponibile in format pdf la inceputul semestrului.
http://thor.info.uaic.ro/~croitoru/ag/ag 13-14 allinone.pdf

Tematica seminariilor

Fiecare seminar dezbate câteva probleme (unele dintre ele dificile !) pentru a aprofunda subiectele introduse la curs. Toate problemele sunt postate la începutul semestrului astfel încât studenții interesați să caute soluții originale sau să studieze probleme similare în bibliografia înrudită.

Bibliografie

- CROITORU C., *Tehnici de bază în optimizarea combinatorie*, Editura Univ. Al. I. Cuza Iasi, Iasi,1992.
- CROITORU C., *Introducere in proiectarea algoritmilor paraleli*, Editura Matrix Rom, Bucuresti, 2002.
- TOMESCU I., *Probleme de combinatorică și teoria grafurilor*, Editura did. și ped., Bucuresti,1981.
- DIESTEL R., *Graph Theory*, Electronic Edition.
- CORMEN T.H., Leiserson C.E., Rivest R.L., Stein C., *Introduction to Algorithms*,MIT Press 2001.

Bibliografie

- CROITORU C., *Tehnici de bază în optimizarea combinatorie*, Editura Univ. Al. I. Cuza Iasi, Iasi,1992.
- CROITORU C., *Introducere in proiectarea algoritmilor paraleli*, Editura Matrix Rom, Bucuresti, 2002.
- TOMESCU I., *Probleme de combinatorică și teoria grafurilor*, Editura did. și ped., Bucuresti,1981.
- DIESTEL R., *Graph Theory*, Electronic Edition.
- CORMEN T.H., Leiserson C.E., Rivest R.L., Stein C., *Introduction to Algorithms*,MIT Press 2001.

Suplimentar

http://thor.info.uaic.ro/~croitoru/ag/resurse bibliografice (optionale)

EVALUARE

Punctajul minim de promovare: 50 puncte.

EVALUARE

Punctajul minim de promovare: 50 puncte.

FORME:

- Activitatea de la seminar (prezența, participare la dezbateri): 0-18 puncte.
- Teme pentru acasă (3 teme, în săptămânile 5, 9,13), maxim 14 puncte fiecare: 0-42 puncte.
- Testul final scris (open books): 0-60 puncte.

EVALUARE

Punctajul minim de promovare: 50 puncte.

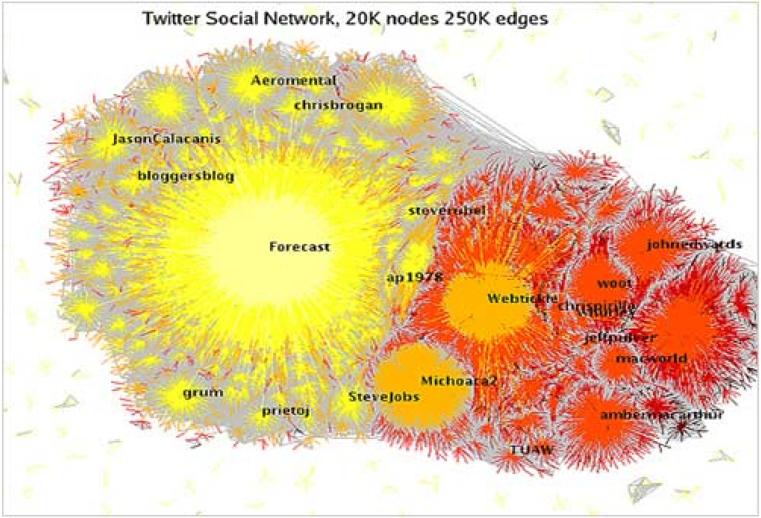
FORME:

- Activitatea de la seminar (prezența, participare la dezbateri): 0-18 puncte.
- Teme pentru acasă (3 teme, în săptămânile 5, 9,13), maxim 14 puncte fiecare: 0-42 puncte.
- Testul final scris (open books): 0-60 puncte.

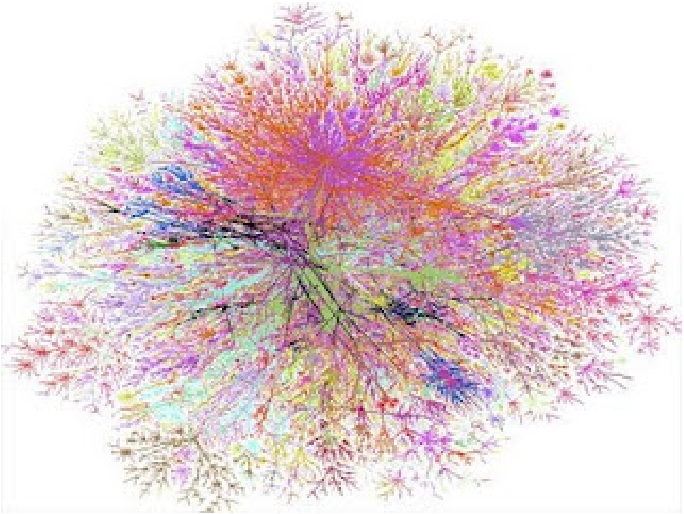
Nota finală

Studenții care au obținut minim 50 puncte, sunt sortați descrescător dupa punctajul final și clasificați dupa regulile ETCS cu adaptările precizate de FII.

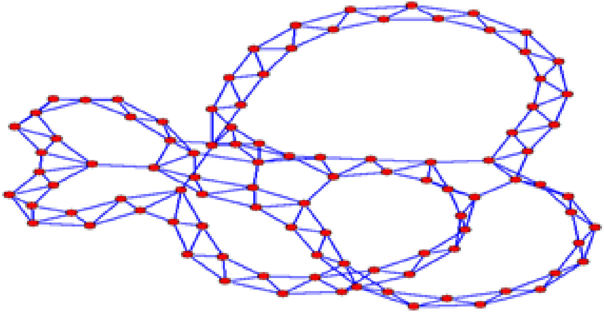
Bonus: Seminar Special.



A nice visualization by Akshay Java of network analysis of Twitter.



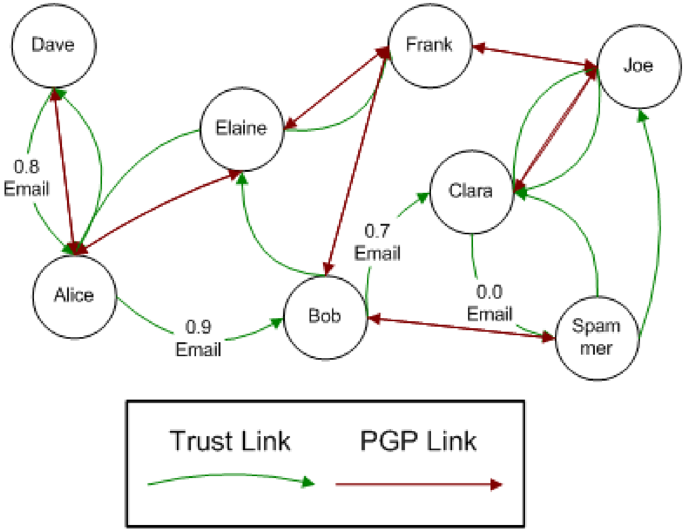
Interest in scale-free networks started in 1999 with work by Albert-László Barabási and colleagues at the University of Notre Dame.



World.png

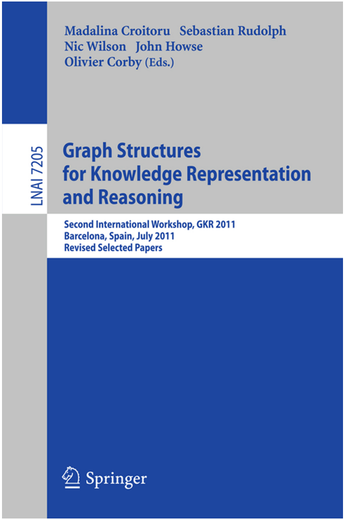
A small-world network is a type of mathematical graph in which most nodes are not neighbors of one another, but most nodes can be reached from every other by a small number of hops or steps.

INTERESUL PENTRU GRAFURI ÎN INFORMATICĂ

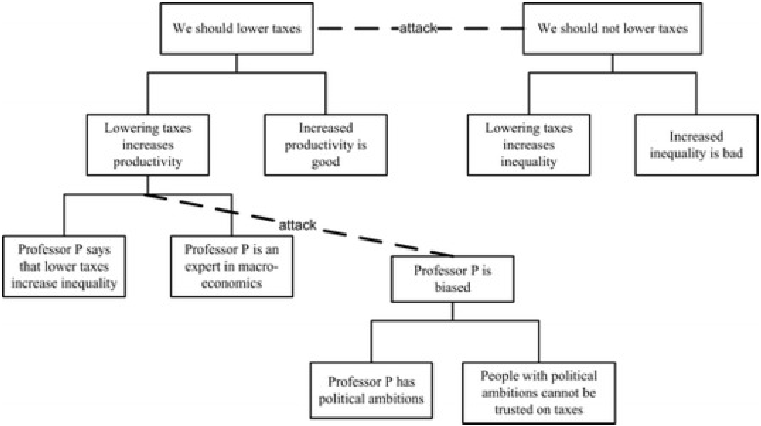


Konfidi – Trust Networks with PGP and RDF.

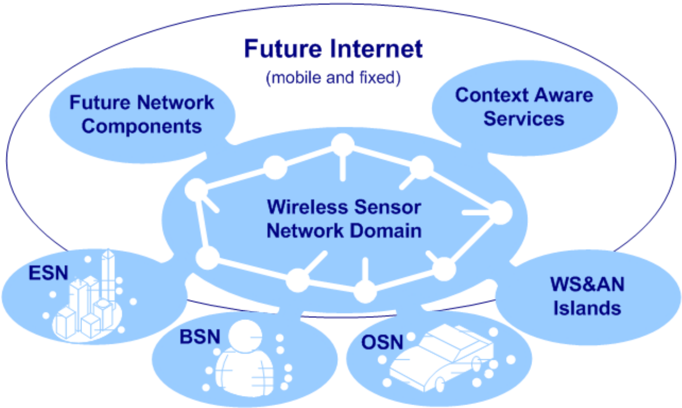
INTERESUL PENTRU GRAFURI ÎN INFORMATICĂ



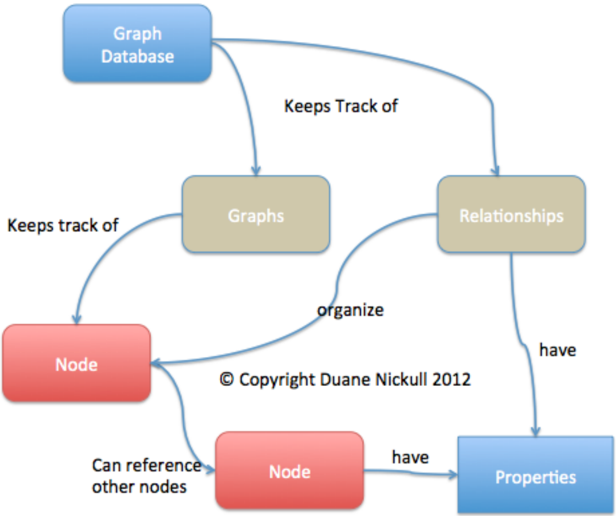
Graph-based knowledge representation formalisms: Bayesian Networks (BNs), Semantic Networks (SNs), Conceptual Graphs (CGs), Formal Concept Analysis (FCA), CP-nets, GAI-nets, etc.



Argumentation Frameworks.

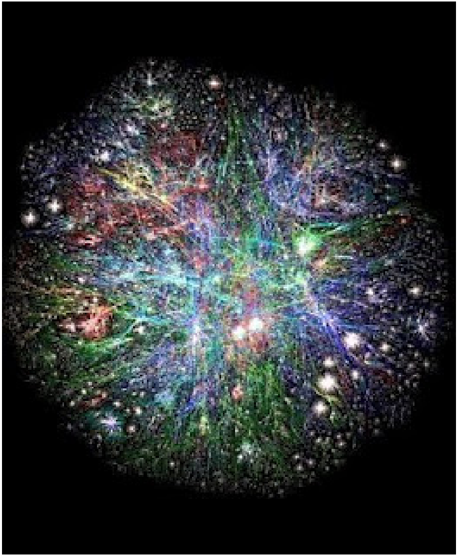


Environmental Sensor Networks (ESN), Object Sensor Networks (OSN) or Body Sensor Network (BSN) operate a variety of different protocols for the specific application environment.



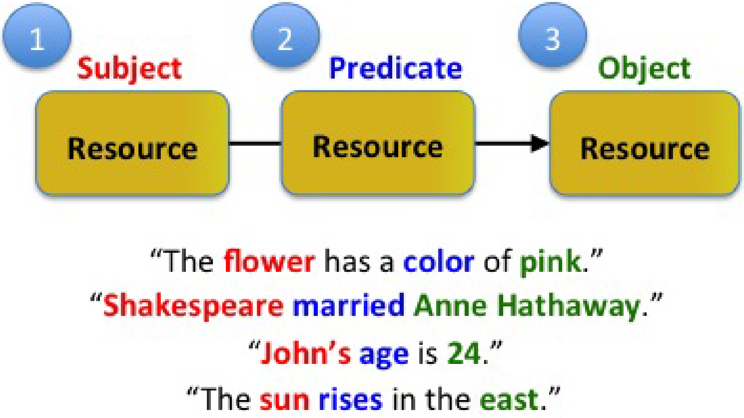
Shot.png

Graph-based Data Basis.

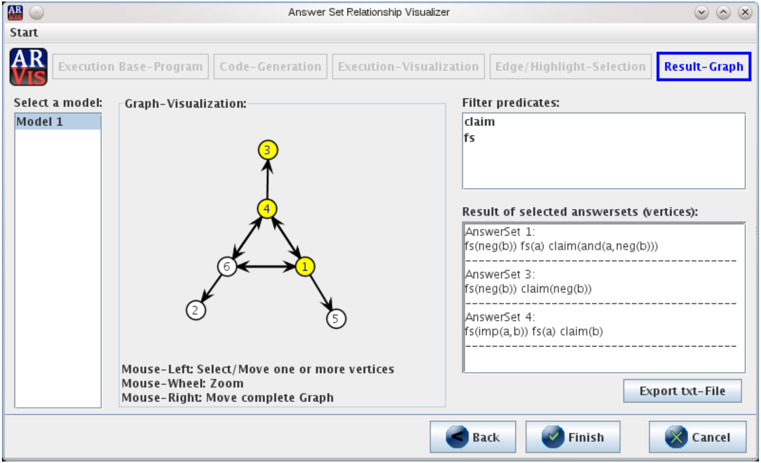


Visualization systems.

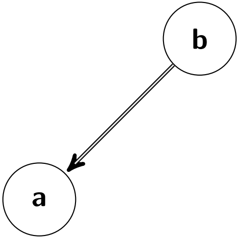
Madrid-Metro.

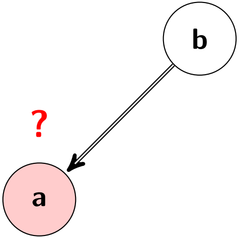


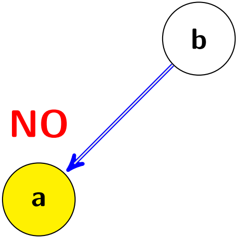
A set of such triples is called an RDF graph.

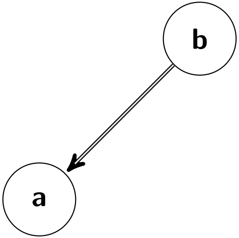


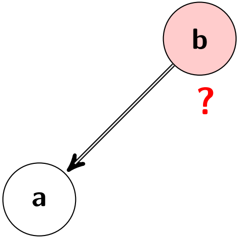
Utilizing ASP for Generating and Visualizing Argumentation Frameworks.

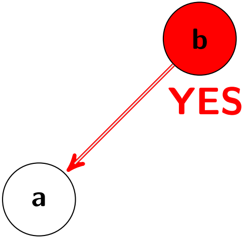

b
a

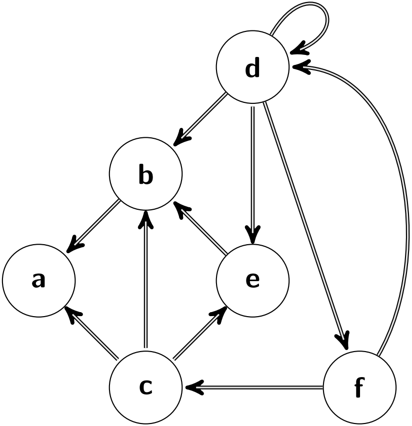

b
?
a

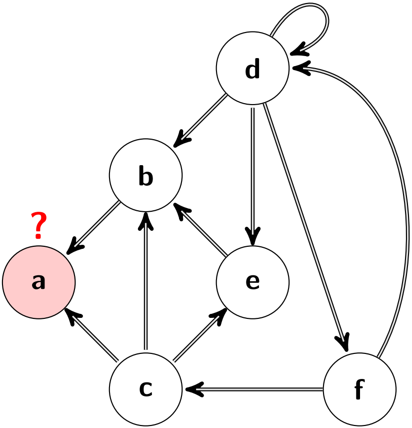

b
NO
a

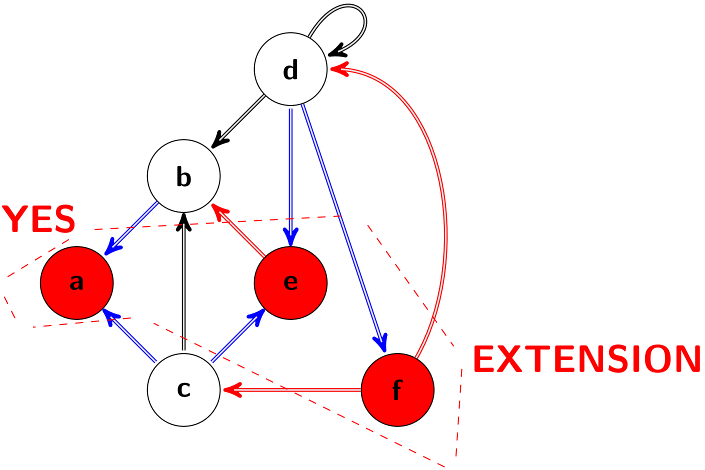

b
a

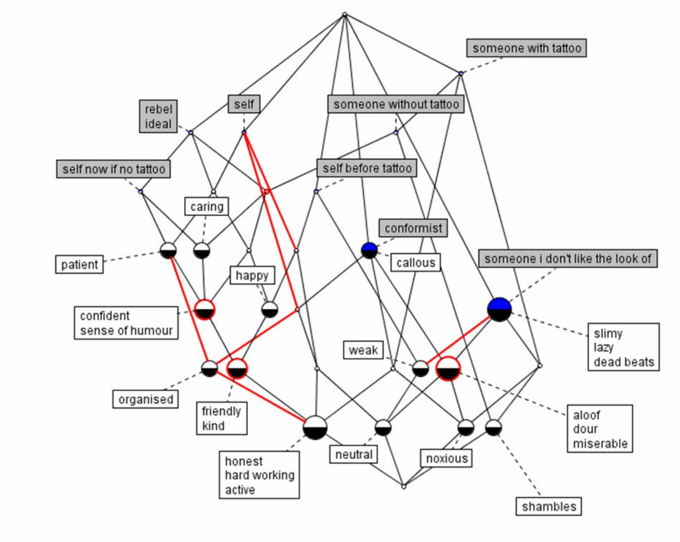

b
?
a


b
YES
a


d
b
a
e
c
f


d
b
?
a
e
c
f


d
b
YES
a
e
c
f
EXTENSION



Visualizing a FCA lattice.

(ag 13-14 allinone.pdf, primul capitol)

P:

Clasa problemelor (de decizie) pentru care exista algoritmi determiniști cu timp polinomial de rezolvare.

NP:

Clasa problemelor (de decizie) pentru care exista algoritmi nedeterminiști cu timp polinomial de rezolvare.

$\mathrm{P} \subseteq \mathrm{NP}$ (Incluziune strictă ?)

Problema P se reduce polinomial la problema Q, dacă orice intrare a problemei P se poate transforma în timp polinomial într-o intrare a problemei Q, astfel încât rezolvând Q pe această intrare se obține răspunsul (corect) pentru P.

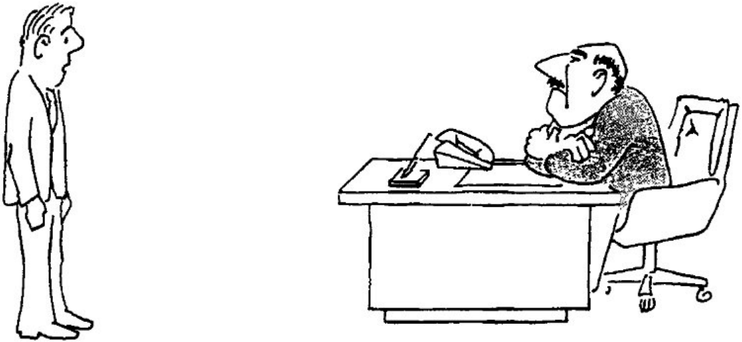
Definiție

Problema de decizie P se numește NP-dificilă (NP-hard) dacă orice problemă din **NP** *se reduce polinomial la P.*

Definiție

Problema de decizie P se numește NP-completă dacă este NP-dificilă și în plus aparține la **NP**.

ELEMENTE INTRODUCTIVE DE COMPLEXITATE



"I can't find an efficient algorithm, I guess I'm just too dumb."

Garey and Johnson, **Computers and Intractability**, 1979.

"I can't find an efficient algorithm, because no such algorithm is possible!"

Garey and Johnson, **Computers and Intractability**, 1979.

ELEMENTE INTRODUCTIVE DE COMPLEXITATE

"I can't find an efficient algorithm, but neither can all these famous people."

Garey and Johnson, **Computers and Intractability**, 1979.

PROBLEMELE PENTRU SEMINARUL 1

1

Fie $a, b \in \mathbf{N}$. Demonstrați că $n^a = O(n^b)$ dacă și numai dacă $a \leq b$. Demonstrați că $n^a = O(e^n)$ și că nu are loc $e^n = O(n^a)$ (e este baza logaritmului natural).

2

Argumentați o evaluare de tipul $T(n) = \Theta(.)$ pentru timpul de executie a algoritmului:

Sumă Triplă (n)

```
s ← 0
for i = 1, n do
   for j = i, n do
      for k = j, n do
         s ← s + 1
```

3

Considerăm următoarele două funcții:

F(n)
if $(n = 1)$ **return** *true*
else return $G(n-1)$

G(n)
if $(n = 1)$ **return** *false*
else return $F(n-1)$

Stabiliți și argumentați valorile $F(2012)$ și $G(2013)$.

3'

Se dispune de un fișier de intrare cu n înregistrări. Prima înregistrare conține numărul n, celelalte $n-1$ conțin fiecare un număr din mulțimea $\{1, 2, \dots, n\}$. Dacă aceste ultime $n-1$ înregistrări conțin numere distincte, rezultă că exact unul dintre numerele $1, 2, \dots, n$ lipsește.

Descrieți un algoritm eficient care să determine numărul lipsă. (n poate fi foarte mare!)

Aveți o soluție și pentru cazul în care lipsesc exact două numere?

4

Pentru înmulțirea a două numere întregi se poate folosi algoritmul descris mai jos prin două exemple. Se observă că operațiile efectuate sunt doar înmulțirea cu doi, împărțirea întreagă la doi și adunarea numerelor întregi.

```
48 × 17                 29 × 135
48     17                 29     135
24     34                 14     270
12     68                  7     540
 6    136                  3    1080
 3    272                  1    2160
 1    544                 ==========
===============               3915
      816
```

(se adună numerele de pe coloana 2 care au pe coloana 1 numere impare)

Scrieți o funcție recursivă pentru produsul a două numere întregi care să corespundă acestui algoritm și demonstrați-i corectitudinea. Stabiliți complexitatea timp $T(n)$ pentru această funcție (n este numărul biților necesari reprezentării binare a fiecăruia dintre cei doi factori).

5

a) Înfășurătoarea convexă a n puncte $P_i(x_i, y_i)$, $i = \overline{1, n}$ din plan, este cel mai mic poligon convex (în raport cu incluziunea) care conține toate cele n puncte. Demonstrați că dacă dispunem de un algoritm care să determine vârfurile înfășurătoarei convexe a n puncte date cu complexitatea timp $T(n)$ atunci putem sorta un vector întreg n-dimensional în timpul $T(n)$.

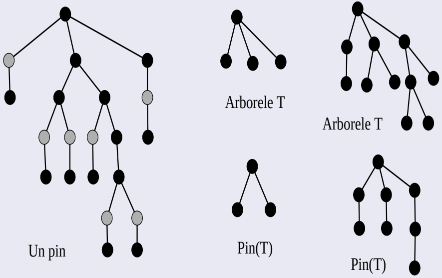
b) Dați două exemple de algoritmi de sortare. Ce complexitate au ?

PROBLEMELE PENTRU SEMINARUL 1

6

Numim *pin* un arbore cu măcar trei noduri cu proprietatea că unicul vecin al oricărei frunze (nod cu un singur vecin) are exact doi vecini. Pentru un arbore T cu cel puțin trei noduri, notăm cu $pin(T)$ subarborele lui T care este pin și are număr maxim de noduri.

Descrieți un algoritm care, pentru T dat, construiește $pin(T)$.



ÎNTREBĂRI ?

Mulțumesc!