

# Programarea algoritmilor eficienti

Alex Popa

Octombrie 2023

## 1 Materiale

1. <http://www.cs.yale.edu/homes/aspnes/classes/202/notes.pdf>
2. *Introduction to Algorithms*: Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein.
3. *Introduction to the Theory of Computation*: Michael Sipser.
4. *Approximation Algorithms*: Vijay V. Vazirani.
5. *Probability and computing*: Michael Mitzenmacher.
6. *Parameterized Complexity Theory*. Jörg Flum and Martin Grohe.

## 2 Modalitatea de notare

**Nota finală = min{10, Nota examen + Nota proiect}**

In examen veti lua o nota intre 0 si 10, iar la proiect veti lua o nota intre 0 si 5. Pentru a promova, nu se cere nota minima in examen sau la proiect. Trebuie doar ca suma lor sa fie peste 5 (usor, nu?).

1. **Examenul** va fi scris, va dura 2 ore si va consta in diverse probleme de algoritmica (discutam mai multe detalii despre structura examenului la ultimul curs). Examenul fa fi fizic, la scoala.
2. **Proiectul** consta in efectuarea unui rezumat al unui articol stiintific din conferintele: STOC, FOCS, SODA, ICALP, ESA, STACS, WAOA, IWOCA, SOFSEM, WADS, SWAT, ISAAC, sau un articol care este varianta de jurnal al unui articol publicat in conferintele anterior mentionate. Puteti alege un articol din alte jurnale/conferinte de algoritmi decat cele mentionate anterior **DOAR** daca aveti aprobarea mea. Puteti cauta articole stiintifice folosind Google sau Google Scholar.

Pana pe **5 noiembrie 2023 ora 23:59** e deadline-ul pana la care trebuie sa va alegeti articolul stiintific si sa il treceti in tabelul:

[https://docs.google.com/spreadsheets/d/1r4p-mePzxKgbabTG5utgwh0g\\_5p95Ljx6BvInubP7ys/edit?usp=sharing](https://docs.google.com/spreadsheets/d/1r4p-mePzxKgbabTG5utgwh0g_5p95Ljx6BvInubP7ys/edit?usp=sharing)

Pana pe **3 decembrie 2023 ora 23:59** trebuie sa trimiteți rezumatul articolului la link-ul: <https://forms.gle/ftUe5siorpKA1Qwz8>. Rezumatul poate fi facut in limba engleza (de preferat, dar nu obligatoriu) sau in limba romana in Latex (de preferat, dar nu obligatoriu) sau in Word. Trebuie sa aveti minim 3 pagini, font Times New Roman de 12, margini de 2cm, spatiere normala intre linii (1). In rezumat trebuie sa descrieti in cuvintele voastre tehnice folosite in articol si sa faceti dovada ca ati inteles cele prezentate in articol. In seminariile incepand cu saptamana 4 decembrie 2023 veti prezenta la seminar articolul ales (in 10-15 minute in functie de cate persoane vor trimite prezentarea).

**ATENTIE:** O intarziere oricat de mica la deadline-ul pentru alegerea articolului **SAU** a trimiterii rezumatului **SAU** lipsa prezentarii implica NOTA ZERO la proiect!!! (Fara discutii)

**ATENTIE (din nou):** Voi testa rezumatele voastre cu soft antiplagiat. Daca aveti paragrafe identice sau foarte asemanatoare, veti primi NOTA ZERO la proiect!

### 3 Programa (tentativa) a cursului

#### 3.1 Introducere

#### 3.2 Clase de complexitate si probleme NP-hard (partea 1)

Capitolul 7: *Introduction to the Theory of Computation*: Michael Sipser.

#### 3.3 Clase de complexitate si probleme NP-hard (partea 2)

Capitolul 7: *Introduction to the Theory of Computation*: Michael Sipser.

#### 3.4 Algoritmi de aproximare (partea 1)

Capitolul 1: *Approximation Algorithms*: Vijay V. Vazirani.

Topicuri:

1. Definitii de baza
2. 2-aproximare pentru problema Vertex Cover

#### 3.5 Algoritmi de aproximare (partea 2)

Capitolul 3: *Approximation Algorithms*: Vijay V. Vazirani. Topicuri: Set cover si Minimum Steiner Tree

#### 3.6 Algoritmi de aproximare (partea 3)

Capitolul 3: *Approximation Algorithms*: Vijay V. Vazirani. Topicuri: Traveling Salesman Problem si Knapsack

#### 3.7 Algoritmi de aproximare (partea 4)

Capitolul 14: *Approximation Algorithms*: Vijay V. Vazirani. Topicuri: Designing approximation algorithms via linear programming

#### 3.8 Algoritmi fixed parameter (partea 1)

Materiale: <http://www.cs.bme.hu/~dmarx/papers/marx-warsaw-fpt1>

Topicuri:

1. Definitii de baza: algoritmi fixed parameter, kernelizare;
2. Exemplificare pentru problema vertex cover si problema de acoperire a unor puncte din plan cu  $k$  drepte

### **3.9 Algoritmi fixed parameter (partea 2)**

Materiale: <http://www.cs.bme.hu/~dmarx/papers/marx-warsaw-fpt1>

Topicuri:

1. Crown decomposition for vertex cover

### **3.10 Algoritmi fixed parameter (partea 3)**

Materiale: <http://www.cs.bme.hu/~dmarx/papers/marx-warsaw-fpt1>

Topicuri studiate:

1. Crown decomposition for the dual of vertex coloring
2. Bounded search trees
3. Hereditary properties
4. Color coding

### **3.11 Algoritmi online (partea 1)**

### **3.12 Algoritmi euristici**

Materiale:

1. Tabu search: <https://slideplayer.com/slide/13277252/>
2. Generalitati despre algoritmi euristici: <https://slideplayer.com/slide/7538887/>
3. Algoritmi genetici: <https://slideplayer.com/slide/4579667/>
4. Gurobi: <https://www.gurobi.com>

### **3.13 Invitat special**

## **4 Seminar**

### **4.1 Probleme de matematici discrete**

Materiale recomandate: <http://www.cs.yale.edu/homes/aspnes/classes/202/notes.pdf>

*Introduction to Algorithms*: Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein.

Subiecte discutate la seminar:

1. Notatiile  $O, o, \omega, \Omega, \Theta$  (Capitolul 3 din Cormen)
2. Demonstratii simple folosind definitiile.
3. Principiul lui Dirichlet / al cutiei / pigeonhole principle.

- 4.2 Probleme NP-hard**
- 4.3 Algoritmi de aproximare**
- 4.4 Algoritmi fixed parameter**
- 4.5 Prezentari proiecte**
- 4.6 Prezentari proiecte**
- 4.7 Prezentari proiecte**

## **5 De ce trebuie sa invatam matematica**

Am am avut o discutie in trecut cu fosti studenti si mi-au spus trei chestii foarte interesante:

1. Ca nu prea intelegeti de ce se face atata matematica in facultate.
2. Ca desi se face o gramada de matematica in liceu si in facultate, nimeni nu va invata cum sa faceti o demonstratie. Se presupune ca ar trebui sa stiti de undeva, dar cam toata lumea sare peste pasul acesta.
3. Sunteti niste persoane care chiar isi doresc din tot sufletul sa invete.

M-am gandit mult la ce am discutat si am decis sa fac un mic pas pentru a va ajuta. Daca intr-adevar e adevarat punctul 3, ceilalti pasi o sa ii faceti dvs.

Pentru a intelege "cum se fac" demonstratii de la zero, va recomand sa cititi un curs de "Discrete Mathematics". Multe din universitatile din vest au publicate online asemenea cursuri si de obicei acestea sunt destinate studentilor din anul 1, tocmai pentru a avea fundamental necesar pentru a studia alte materii din facultatea de matematica/informatica. Deci pur si simplu trebuie sa cautati pe google "discrete mathematics lecture notes" si o sa gasiti o varietate de surse de informare. Dupa cum am spus, aceste cursuri nu necesita cunostinte prelimiare, deci trebuie doar sa aveti vointa.

Mai concret, va trimit un link catre un asemenea curs de la Yale:  
<http://www.cs.yale.edu/homes/aspnes/classes/202/notes.pdf>

Si, ca sa fac legatura cu titlul mailului, in documentul pe care vi l-am trimis, o sa gasiti raspunsul la urmatoarele doua intrebari:

1. So why do I need to learn all this nasty mathematics?
2. But isn't math hard?

Asadar ma astept ca toata lumea sa citeasca (cel putin) aceste doua capitole.