



Anatol GREMALSCHI

doctor habilitat, profesor universitar, UTM

## Instruirea de performanță la Informatică în contextul curriculumului general

**Rezumat:** Perfecționarea permanentă a procesului de instruire în domeniul informaticii școlare la nivel avansat, solicită elaborarea unui document reglator – curriculumul de performanță pentru disciplina Informatică. Prin acest articol inițiem o discuție largă asupra proiectului de curriculum, elaborat de noi în baza experienței avansate și a analizei documentelor normative naționale și internaționale.

**Abstract:** Systemic development of the performance training process in the field of computer science at the national level requires the elaboration of a regulating document in the field – Performance curriculum for Computer science. This paper proposes for a wide discussion a project of national performance curriculum for Computer science, developed on base of the analysis of national and international normative documents and experience.

**Abstract:** Systemic development of the performance training process in the field of computer science at the national level requires the elaboration of a regulating document in the field – Performance curriculum for Computer science. This paper proposes for a wide discussion a project of national performance curriculum for Computer science, developed on base of the analysis of national and international normative documents and experience.

Analiza comparativă a curriculumului național la Informatică și a curriculumului de performanță în domeniul informaticii efectuată anterior [1, p. 46] demonstrează necesitatea elaborării unui document național, care ar pune începuturile unei abordări sistemice a activităților didactice în acest domeniu. Problema principală cu care se confruntă elaboratorii de curriculum destinat formării de performanțe constă, în primul rând, în identificarea competențelor ce ar pune în valoare întreg potențialul creativ al elevilor.

Scopul acestui articol constă în prezentarea unui proiect de curriculum, bazat pe experiența națională și cea internațională, care ar servi drept suport metodologic pentru cadrele didactice care lucrează cu elevii ce manifestă interes pentru atingerea unor rezultate de performanță în domeniul informaticii școlare.



Sergiu CORLAT

lector superior, Universitatea de Stat din Tiraspol

Analiza comparată a unui set de curriculumuri naționale (Moldova, România, Rusia, Bulgaria, Belarus etc.) și a proiectelor de curriculumuri internaționale, elaborate de Comitetul Științific al Olimpiadelor Internaționale la Informatică (2005-2010), demonstrează faptul că în cazul elevilor de performanță un curriculum avansat ar trebuie să includă următoarele domenii de studii:

• aritmetica și geometria de calculator;  
• mulțimi, relații și funcții;  
• logica matematică;  
• tehnicile de demonstrare;  
• combinatorica;  
• grafuri și arbori;  
• structuri fundamentale de date;  
• elemente avansate de programare;  
• analiza complexității algoritmilor;  
• algoritmi fundamentali și strategiile algoritmice.

- aritmetica și geometria de calculator;
- mulțimi, relații și funcții;
- logica matematică;
- tehnicile de demonstrare;
- combinatorica;
- grafuri și arbori;
- structuri fundamentale de date;
- elemente avansate de programare;
- analiza complexității algoritmilor;
- algoritmi fundamentali și strategiile algoritmice.

Pornind de la aceste premise, prezentăm o listă de subcompetențe care pot fi dezvoltate prin studierea domeniilor enumerate mai sus, dar care nu se regăsesc în curriculumul național la disciplinele Informatică și Matematică.

Subcompetențe	Conținuturi	Activități educaționale
<b>Grafuri și Arbori</b>		
<ul style="list-style-type: none"> <li>Să cunoască noțiunile de bază ale teoriei grafurilor și metodele de reprezentare a grafurilor în calculator;</li> <li>să descrie în pseudocod algoritmi de parcurgere a grafurilor, de calcul a: drumului minim, arborilor de cost minim, centrului, medianei, componentelor tare conexe, fluxului maxim, mulțimilor maxim independente, ciclurilor Euler și Hamilton;</li> <li>să elaboreze într-un limbaj de nivel înalt programe pentru implementarea algoritmilor în studiu;</li> <li>să identifice probleme din viața reală care pot fi soluționate cu ajutorul algoritmilor în studiu;</li> <li>să determine complexitatea temporală a algoritmilor implementați.</li> </ul>	<p><i>Noțiuni:</i></p> <ul style="list-style-type: none"> <li>graf (conex, planar, orientat, bipartit);</li> <li>vîrf (sursă, destinație).</li> <li>muchie (arc);</li> <li>cale (drum, lanț);</li> <li>ciclu (euler, hamilton);</li> <li>grad (interior, exterior).</li> </ul> <p><i>Reprezentări:</i></p> <ul style="list-style-type: none"> <li>matrice de incidență;</li> <li>matrice de adiacență;</li> <li>listă de vîrfuri;</li> <li>listă de muchii.</li> </ul> <p><i>Parcurgeri:</i></p> <ul style="list-style-type: none"> <li>în adîncime;</li> <li>în lățime.</li> </ul> <p><i>Componente tare conexe:</i></p> <ul style="list-style-type: none"> <li>algoritmul Kosaraju.</li> </ul> <p><i>Arbori de cost minim:</i></p> <ul style="list-style-type: none"> <li>algoritmul Kruskal;</li> <li>algoritmul Prim.</li> </ul> <p><i>Drumul minim:</i></p> <ul style="list-style-type: none"> <li>algoritmul Dijkstra;</li> <li>algoritmul Floyd;</li> <li>mediana;</li> <li>centrul grafului.</li> </ul> <p><i>Flux maxim:</i></p> <ul style="list-style-type: none"> <li>algoritmul Ford Fulkerson.</li> </ul>	<p>Rezolvarea problemelor-tip care necesită implementarea algoritmilor în studiu:</p> <ul style="list-style-type: none"> <li>problema liderului;</li> <li>problema podurilor din Königsberg;</li> <li>problema comisului voiajor;</li> <li>problema poștașului chinez;</li> <li>problema amplasării optime a centrului de deservire (depozitelor);</li> <li>problema minimizării lungimii rețelei;</li> <li>problema distanței minime între două puncte pe graf;</li> <li>problema repartizării optime a resurselor etc.</li> </ul> <p>Rezolvarea și evaluarea problemelor pe platformele de instruire de performanță online.</p>
<b>Structuri fundamentale de date</b>		
<ul style="list-style-type: none"> <li>Să identifice structurile de date în studiu;</li> <li>să descrie proprietățile și operațiile caracteristice fiecărei structuri de date în studiu;</li> <li>să implementeze operațiile caracteristice structurilor de date în studiu într-un limbaj de programare de nivel înalt;</li> <li>să identifice probleme din viața reală, a căror rezolvare necesită utilizarea structurilor de date în studiu.</li> </ul>	<p><i>Arbori:</i></p> <ul style="list-style-type: none"> <li>arbori binari de căutare;</li> <li>arbori roșu-negru;</li> <li>heap-uri;</li> <li>arbori de intervale;</li> <li>arbori fenwick;</li> <li>hash;</li> <li>arbore de sufixări.</li> </ul>	<p>Implementarea operațiilor admise pentru structurile în studiu:</p> <ul style="list-style-type: none"> <li>inițializarea structurilor;</li> <li>adăugarea și lichidarea elementelor;</li> <li>căutarea;</li> <li>parcurgerea.</li> </ul> <p>Rezolvarea problemelor-tip care necesită utilizarea structurilor de date în studiu:</p> <ul style="list-style-type: none"> <li>determinarea înălțimii unui arbore binar;</li> <li>reprezentarea mulțimilor disjuncte;</li> <li>construirea arborilor de compresie Huffman;</li> <li>construirea arborilor asociați expresiilor aritmetice etc.</li> </ul> <p>Rezolvarea și evaluarea problemelor pe platformele de instruire de performanță online.</p>

Subcompetențe	Conținuturi	Activități educaționale
<b>Strategii algoritmice</b>		
<ul style="list-style-type: none"> <li>Să descrie caracteristicile specifice fiecărei strategii de programare;</li> <li>să determine complexitatea algoritmilor implementați cu ajutorul strategiilor algoritmice în studiu;</li> <li>să selecteze strategia optimă pentru rezolvarea problemelor informatice.</li> </ul>	<p><i>Tehnici avansate de programare:</i></p> <ul style="list-style-type: none"> <li>metoda Branch and bound;</li> <li>metoda Divide et Impera, utilizare avansată;</li> <li>programarea dinamică (descriere generală, spre sfârșit, spre început, probleme-tip)</li> </ul>	<p>Rezolvarea problemelor-tip care necesită aplicarea strategiilor în studiu:</p> <ul style="list-style-type: none"> <li>Turnurile din Hanoi;</li> <li>înmulțirea eficientă a numerelor;</li> <li>înmulțirea eficientă a matricelor;</li> <li>problema rucsacului;</li> <li>determinarea centrului absolut pe graf;</li> <li>transformări rapide Fourier;</li> <li>arborii Quad;</li> <li>acumularea punctajului maxim;</li> <li>traectorii distincte de deplasare (linie, plan, spațiu);</li> <li>distribuirea cadourilor.</li> </ul> <p>Rezolvarea și evaluarea problemelor pe platformele de instruire de performanță online.</p>
<b>Algoritmi fundamentali</b>		
<ul style="list-style-type: none"> <li>Să selecteze structurile de date adecvate pentru implementarea operațiilor cu numere lungi;</li> <li>să implementeze operațiile cu numere lungi într-un limbaj de programare de nivel înalt;</li> <li>să genereze elementele combinatorice primare și avansate;</li> <li>să implementeze cu ajutorul algoritmilor într-un limbaj de programare de nivel înalt generarea elementelor combinatorice primare și avansate;</li> <li>să genereze submulțimile și partițiile mulțimilor;</li> <li>să implementeze cu ajutorul algoritmilor într-un limbaj de programare de nivel înalt generarea submulțimilor și partițiilor cu proprietăți date;</li> <li>să cunoască și să implementeze algoritmi eficienți de căutare și sortare;</li> <li>să identifice problemele a căror rezolvare necesită utilizarea algoritmilor în studiu.</li> </ul>	<p><i>Operații cu numere lungi:</i></p> <ul style="list-style-type: none"> <li>adunarea;</li> <li>scăderea;</li> <li>înmulțirea;</li> <li>împărțirea.</li> </ul> <p><i>Algoritmi combinatorici:</i></p> <ul style="list-style-type: none"> <li>permutări și combinări;</li> <li>poziția elementelor;</li> <li>determinarea indicelui elementului;</li> <li>elemente combinatorice „vecine”;</li> <li>numere remarcabile.</li> </ul> <p><i>Algoritmi cu mulțimi:</i></p> <ul style="list-style-type: none"> <li>submulțimi;</li> <li>submulțimi cu indicele dat;</li> <li>indicele submulțimii;</li> <li>submulțimi „vecine”;</li> <li>partiții.</li> </ul> <p><i>Algoritmi de căutare:</i></p> <ul style="list-style-type: none"> <li>căutare liniară;</li> <li>căutare binară.</li> </ul> <p><i>Algoritmi eficienți de sortare:</i></p> <ul style="list-style-type: none"> <li>sortare rapidă;</li> <li>heap sort;</li> <li>merge sort.</li> </ul>	<p>Rezolvarea problemelor-tip care necesită aplicarea algoritmilor în studiu:</p> <ul style="list-style-type: none"> <li>adunarea și scăderea numerelor lungi;</li> <li>înmulțirea eficientă a numerelor lungi;</li> <li>generarea elementelor combinatorice primare (aranjamente, amplasări, permutări, combinări);</li> <li>generarea elementelor combinatorice avansate (numere Catalan, numere Stirling, numere Bell);</li> <li>generarea submulțimilor unei mulțimi, a submulțimilor cu proprietăți date;</li> <li>problema „ghicirii” numărului;</li> <li>gestionarea structurilor de tipul heap (inserare, eliminare);</li> <li>compararea metodelor eficiente de sortare etc.</li> </ul> <p>Rezolvarea și evaluarea problemelor pe platformele de instruire de performanță online.</p>

Subcompetențe	Conținuturi	Activități educaționale
<b>Analiza avansată a algoritmilor</b>		
<ul style="list-style-type: none"> <li>Să descrie algoritmi numerici în studiu;</li> <li>să implementeze într-un limbaj de programare de nivel înalt algoritmi numerici în studiu;</li> <li>să identifice problemele a căror rezolvare eficientă necesită utilizarea algoritmilor numerici în studiu;</li> <li>să descrie algoritmi în studiu pentru prelucrarea șirurilor de caractere;</li> <li>să identifice șirurile de caractere ciclice și periodice;</li> <li>să implementeze într-un limbaj de programare de nivel înalt algoritmi în studiu pentru prelucrarea șirurilor de caractere;</li> <li>să identifice problemele a căror rezolvare eficientă necesită utilizarea algoritmilor în studiu pentru prelucrarea șirurilor de caractere;</li> <li>să descrie algoritmi în studiu pentru rezolvarea problemelor complexe pe grafuri;</li> <li>să identifice centrele și medianele în graf, mulțimile maxim independente și cuplajele maxime;</li> <li>să implementeze într-un limbaj de programare de nivel înalt algoritmi în studiu pentru rezolvarea problemelor complexe pe grafuri;</li> <li>să identifice problemele a căror rezolvare eficientă necesită utilizarea algoritmilor pentru rezolvarea problemelor complexe pe grafuri;</li> <li>să combine eficient diferiți algoritmi pentru rezolvarea problemelor complexe.</li> </ul>	<p><i>Algoritmi numerici:</i></p> <ul style="list-style-type: none"> <li>descompunerea în factori primi;</li> <li>ciurul lui Erastofene;</li> <li>algoritmul Euclid;</li> <li>metoda Gauss pentru rezolvarea sistemelor de ecuații;</li> <li>algoritmi eficienți pentru ridicarea la putere;</li> <li>algoritmi eficienți pentru verificarea numerelor prime;</li> <li>logaritmare eficientă;</li> <li>extragerea eficientă a rădăcinilor;</li> <li>descompunerea eficientă în factori primi.</li> </ul> <p><i>Algoritmi pe șiruri de caractere:</i></p> <ul style="list-style-type: none"> <li>căutarea subșirului în șir: metode brute, algoritmul Knut-Morris-Pratt;</li> <li>șiruri periodice și ciclice;</li> <li>ajustarea optimă a șirului;</li> <li>algoritmul Bayer-Moore;</li> <li>generarea arborelui de sufixe;</li> <li>descompunerea în șiruri simple (prime);</li> <li>algoritmul Aho-Corasick.</li> </ul> <p><i>Algoritmi pe grafuri:</i></p> <ul style="list-style-type: none"> <li>mulțimea maxim independentă;</li> <li>cuplajul maxim (algoritmul de forță brută, algoritmul maghiar);</li> <li>colorarea grafului;</li> <li>P-centre;</li> <li>P-mediane.</li> </ul>	<p>Rezolvarea problemelor:</p> <ul style="list-style-type: none"> <li>calcularea factorialului pentru <math>n &gt; 10</math>;</li> <li>calcularea numărului de zerouri finale în factorial;</li> <li>calcularea factorialului în baza <math>k</math>;</li> <li>calcularea numărului de zerouri finale în factorialul în baza <math>k</math>;</li> <li>determinarea cifrei de pe poziția <math>k</math> în <math>n!</math>;</li> <li>problema de maximizare (minimizare) a funcției într-un set de restricții date;</li> <li>șiruri palindromice și transformarea lor;</li> <li>editarea șirurilor;</li> <li>subșirul maxim comun a două șiruri;</li> <li>subșirul maxim comun a <math>n</math> șiruri;</li> <li>transformări ale șirurilor periodice;</li> <li>identificarea elementelor în șiruri periodice;</li> <li>transformări ale structurilor ciclice;</li> <li>identificarea elementelor în structurile ciclice;</li> <li>descompunerea șirurilor;</li> <li>identificarea elementelor în șiruri cu proprietăți date;</li> <li>acoperirea cu un număr minim (maxim) de elemente;</li> <li>realizarea paralelă a proiectelor;</li> <li>amplasarea optimă a <math>n</math> centre de deservire etc.</li> </ul> <p>Rezolvarea și evaluarea problemelor pe platformele de instruire de performanță online.</p>
<b>Algoritmi geometrici</b>		
<ul style="list-style-type: none"> <li>Să cunoască noțiunile fundamentale ale geometriei computaționale;</li> <li>să descrie în pseudocod algoritmi fundamentali de geometrie computațională: intersecția dreptelor (drepte și segmentului, segmentelor) în plan; poziționarea punctului față de vector, transformări de coordonate, înfășurătoarea convexă, aria poligonului simplu, apartenența punctului la domeniu, cea mai apropiată pereche de puncte, nucleul poligonului, triangularizări, apropiere;</li> <li>să elaboreze într-un limbaj de nivel înalt programe pentru implementarea algoritmilor în studiu;</li> </ul>	<p><i>Noțiuni fundamentale:</i></p> <ul style="list-style-type: none"> <li>punct;</li> <li>segment;</li> <li>dreaptă;</li> <li>plan;</li> <li>sistem de coordonate.</li> </ul> <p><i>Transformări de coordonate:</i></p> <ul style="list-style-type: none"> <li>deplasări;</li> <li>rotații;</li> <li>transformări mixte;</li> <li>coordonate polare.</li> </ul> <p><i>Intersecții:</i></p> <ul style="list-style-type: none"> <li>coordonate omogene;</li> <li>intersecția a două drepte;</li> <li>intersecția a două segmente;</li> <li>intersecția segmentului și a dreptei.</li> </ul>	<p>Rezolvarea problemelor-tip care necesită implementarea algoritmilor în studiu:</p> <ul style="list-style-type: none"> <li>problema cadrării;</li> <li>problema calculării ariei unui domeniu;</li> <li>problema fugarului;</li> <li>problema poligonului și a diagramei Voronoi;</li> <li>problema controlului teritoriului;</li> <li>problema divizării figurilor plane în triunghiuri;</li> <li>problema triangularizării Delone;</li> <li>problema domeniului convex de arie minimă care va conține o mulțime dată etc.</li> </ul> <p>Rezolvarea și evaluarea problemelor pe platformele de instruire de performanță online.</p>

Subcompetențe	Conținuturi	Activități educaționale
<ul style="list-style-type: none"> <li>să identifice probleme din viața reală care pot fi soluționate cu ajutorul algoritmilor în studiu;</li> <li>să determine complexitatea temporală a algoritmilor implementați.</li> </ul>	<p><i>Înfășurătoarea convexă:</i></p> <ul style="list-style-type: none"> <li>algoritmii de forță brută;</li> <li>algoritmii Graham;</li> <li>algoritmii Andrew.</li> </ul> <p><i>Aria poligonului simplu:</i></p> <ul style="list-style-type: none"> <li>triangularizarea;</li> <li>metoda trapezelor.</li> </ul> <p><i>Algoritmi fundamentali:</i></p> <ul style="list-style-type: none"> <li>cea mai apropiată pereche de puncte;</li> <li>triangularizarea poligonului simplu;</li> <li>determinarea nucleului poligonului;</li> <li>domenii cu proprietăți prestabilite.</li> </ul>	

Întru realizarea eficientă a activităților educaționale enumerate mai sus, inserăm o listă de publicații și adresele câtorva platforme Web pentru instruire de performanță.

Invităm cadrele didactice ce predau disciplina *Informatica* să își expună opinia față de acest proiect de curriculum, accesînd pagina Web <http://informaticadeconcurs.wordpress.com/sondaj/>.

#### Literatura recomandată:

##### Ediții în limba română:

- Cerchez, Em., *Programarea în limbajul C/C++ pentru liceu*, vol. I-III, Ed. Polirom, Iași, 2007.
- Cormen, Th., *Introducere în algoritmi*, Ed. Agora, Cluj, 2001.
- Corlat, S., *Algoritmi și probleme de geometrie computațională*, Ed. Prut Internațional, Chișinău, 2009.
- Giumale, Cr., *Introducere în analiza algoritmilor*, Ed. Polirom, Iași, 2004.
- Logofătu, D., *Algoritmi fundamentali în C. Aplicații*, Ed. Polirom, Iași, 2007.
- Oltean, M., *Programarea și implementarea algoritmilor*, Ed. Agora, Cluj, 1999.
- Oltean, M., *Programarea jocurilor matematice*, Ed. Albastră, Cluj, 1996.

##### Ediții în limba rusă:

- Новиков, Ф.А., *Дискретная математика для программистов*, Изд. Питер, Санкт Петербург, 2001.
- Пападимитриу, Х., *Комбинаторная оптимизация. Алгоритмы и сложность*, Изд. Мир, Москва, 1985.
- Кристофидес, П., *Теория графов. Алгоритмический подход*, 1978, Изд. Мир, Москва.
- Алферов, А., *Основы криптографии*, Изд. Гелиос, Москва, 2002.

- Стенли, Р., *Перечислительная комбинаторика*, Изд. Мир, Москва, 1990.
- Нагао, М., *Структуры и базы данных*, Изд. Мир, Москва, 1986.
- Ласло, М., *Вычислительная геометрия и компьютерная графика на C++*, Изд. Бином, Москва, 1997.

##### Ediții în limba engleză:

- Sedgewick Th., *Algorithms in C*, Addison Wesley, 2001.
- Gibbons, A., *Algorithmic graph theory*, Addison Wesley, 1999.
- Goldberg, D., *Genetic Algorithms*, Addison Wesley, 2003.
- Williamson, G., *Combinatorics for Computer Science*, Dover publications, New York, 2002.

##### Platforme Web de instruire:

- Infoarena.ro
- Campion.edu.ro
- Informatics.ru
- Usaco.org

#### REFERINȚE BIBLIOGRAFICE:

- Gremalschi, A.; Corlat, S., *Rolul interdisciplinarității Matematică-Informatică în pregătirea elevilor de performanță*, în *Didactica Pro...*, nr.4 (68), 2011.
- Кирюхин, В.М., *Информатика. Международные олимпиады*, Изд. Просвещение, Москва, 2009.
- Кирюхин, В.М., *Методика проведения и подготовки к участию в олимпиадах по информатике*, Изд. Бином, Москва, 2011.
- Informatica. Curriculum pentru învățământul liceal*, Ed. Știința, Chișinău, 2010.
- The international Olympiad in Informatics Syllabus*, <http://www.ioi2009.org/GetResource?id=32>