

GÂNDIREA ALGORITMICĂ - PUNTE ÎNTRE MATEMATICĂ ȘI INFORMATICĂ ÎN FORMAREA COMPETENȚELOR ELEVILOR DE LICEU

Corina Curilă and Ioana Alina Luncan

Abstract. Gândirea algoritmică, componentă esențială a gândirii computaționale, se regăsește în mod natural în procesele cognitive implicate în învățarea matematicii. Articolul de față propune o analiză teoretică a modului în care principiile algoritmice pot contribui la dezvoltarea competențelor logice și de rezolvare de probleme la elevii de liceu. Sunt discutate analogiile dintre raționamentul matematic și cel algoritmic, precum și modalități de integrare a gândirii algoritmice în activitățile de predare a matematicii. Exemplele propuse evidențiază potențialul acestei abordări pentru a spori claritatea gândirii, autonomia intelectuală și transferul de competențe între disciplinele matematică și informatică.

MSC 2000.

Key words. gândire algoritmică, didactica matematicii, interdisciplinaritate, gândire computațională, liceu.

1. INTRODUCERE

În contextul actual al educației STEM, integrarea dintre matematică și informatică devine din ce în ce mai relevantă. Ambele discipline promovează gândirea logică, abstractizarea și structura raționamentului. Totuși, în practica școlară, aceste domenii sunt adesea tratate separat, ceea ce limitează potențialul formativ al fiecăruia. Conceptul de gândire algoritmică oferă un cadru comun care poate unifica experiențele de învățare din cele două domenii. Termenul face referire la capacitatea de a formula, descrie și aplica proceduri sistematice (algoritmi) pentru rezolvarea unei probleme. Într-un sens extins, presupune nu doar scrierea de algoritmi în limbaje de programare, ci și abordarea logică, pas cu pas, a oricărei sarcini intelectuale (Wing, 2006).

De asemenea, în lucrarea "Mathematical Structures in Computer Science" sunt prezentate principii ale aplicării matematicii în implementarea algoritmilor, importante în utilizarea în practică a noțiunilor teoretice predate de profesor la clasă (Gersting, 1990).

2. FUNDAMENT TEORETIC: CONEXIUNI ÎNTRE GÂNDIREA ALGORITMICĂ ȘI GÂNDIREA MATEMATICĂ

Gândirea matematică și cea algoritmică împărtășesc numeroase trăsături: ambele implică analiză, modelare, verificare și generalizare. Algoritmul fiind cel care, folosind datele de intrare, utilizează o succesiune de pași succinți, clari și corecți, oferind ca date de ieșire soluția problemei abordate.

În matematică, elevii sunt învățați să urmeze raționamente logice pentru a ajunge la o concluzie validă - exact cum un algoritm parcurge pași definiți pentru a obține un rezultat. Prin urmare, gândirea algoritmică poate fi privită ca un instrument meta-cognitiv pentru înțelegerea și formalizarea raționamentului matematic.

Elevul care învață să gândească algoritmic devine mai conștient de propriul proces de rezolvare.

3. INTEGRAREA GÂNDIRII ALGORITMICE ÎN PREDAREA MATEMATICII LA LICEU

Integrarea gândirii algoritmice nu presupune transformarea lecțiilor de matematică în lecții de programare, ci mai degrabă reformularea modului în care sunt abordate problemele și conceptele. Profesorul poate introduce treptat elemente algoritmice în activitățile obișnuite, cum ar fi:

Reformularea problemelor în termeni algoritmici

Exemplu (clasa a IX-a): "Scrie un algoritm care determină soluțiile unei ecuații de forma $ax + b = 0$."

```
Citeste a, b
Daca a ≠ 0 atunci
    x ← -b / a
    Afiseaza "Solutia este ", x
Altfel
    Daca b = 0 atunci
        Afiseaza "Ecuatie adevarata pentru orice x"
    Altfel
        Afiseaza "Ecuatie imposibila"
SfarsitDaca
```

Determinarea divizorilor unui număr

Exemplu (clasa a IX-a - lecția despre divizibilitate): Elevii pot fi provocați să scrie un algoritm care determină toți divizorii unui număr natural și să observe că aceștia apar în perechi complementare, iar în cazul pătratelor perfecte, un divizor se repetă o singură dată.

```
Citeste n
Pentru i ← 1 pana la radical(n) executa
    Daca n % i = 0 atunci
        Daca i = n / i atunci
            Afiseaza i // cazul patrat perfect
        Altfel
            Afiseaza i si n / i
    SfarsitDaca
SfarsitPentru
```

Determinarea dacă un număr este prim

Exemplu (clasa a IX-a - proprietăți ale numerelor naturale): Elevii pot verifica dacă un număr este prim testând divizibilitatea până la rădăcina pătrată a numărului.

```

Citeste n
Daca n < 2 atunci
    Afiseaza "Nu este prim"
Opreste
prim ← adevarat
Pentru i ← 2 pana la radical(n) executa
    Daca n % i = 0 atunci
        prim ← fals
SfarsitPentru
Daca prim atunci
    Afiseaza "Numarul este prim"
Altfel
    Afiseaza "Numarul nu este prim"

```

Descompunerea unui număr în factori primi

Exemplu (clasa a IX-a - descompunerea în factori): Elevii pot descrie pas cu pas procesul de împărțire succesivă pentru a determina factorii primi ai unui număr.

```

Citeste n
p ← 2
Cat timp n > 1 executa
    Daca n % p = 0 atunci
        Afiseaza p
        n ← n / p
    Altfel
        p ← p + 1
SfarsitCatTimp

```

Determinarea numerelor prime până la un număr dat (Ciurul lui Eratostene)

Exemplu (clasa a X-a - recapitulare despre numere prime și compuse): Elevii pot descrie pașii algoritmului lui Eratostene și observa că este un proces iterativ de eliminare logică.

```

Citeste n
Marcheaza toate numerele de la 2 la n ca "prime"
Pentru i ← 2 pana la radical(n) executa
    Daca i este marcat "prim" atunci
        Pentru j ← i*i pana la n cu pas i executa

```

```

    Marcheaza j ca "neprim"
  SfarsitPentru
SfarsitPentru
Afiseaza toate numerele marcate "prime"

```

Utilizarea pseudocodului sau a diagramelor de flux

Exemplu (clasa a X-a): Într-o lecție despre progresii aritmetice, elevii pot fi invitați să scrie un pseudocod care calculează termenul n al unei progresii:

```

Citeste a1, r, n
an ← a1 + (n - 1) * r
Afiseaza an

```

Explorarea proceselor recursive

Recursivitatea este un concept comun ambelor discipline. Exemplu (clasa a XI-a): Definirea recursivă a șirului lui Fibonacci: $F_1 = 1$, $F_2 = 1$, $F_n = F_{n-1} + F_{n-2}$.

```

Funcție Fibonacci(n)
  Dacă n = 1 sau n = 2 atunci
    returnează 1
  Altfel
    returnează Fibonacci(n-1) + Fibonacci(n-2)
SfarsitFuncție

```

4. BENEFICII ȘI PROVOCĂRI

Introducerea gândirii algoritmice în predarea matematicii aduce multiple beneficii:

- crește claritatea raționamentului elevilor;
- consolidează înțelegerea proceselor logice;
- facilitează transferul de cunoștințe între matematică și informatică;
- stimulează autonomia și creativitatea intelectuală.

Totuși, aplicarea acestei abordări necesită o bună pregătire a profesorilor și o colaborare reală între cadrele didactice de matematică și informatică.

5. CONCLUZII

Gândirea algoritmică reprezintă o punte autentică între matematică și informatică, oferind un limbaj comun pentru exprimarea logicii și a raționamentului. În liceu, integrarea acestei perspective permite o înțelegere mai profundă a matematicii, transformând procesul de învățare într-o explorare activă și rațională. Formarea gândirii algoritmice nu se reduce la scrierea de programe, ci presupune dezvoltarea unei atitudini cognitive orientate spre analiză, planificare și verificare - competențe esențiale pentru învățarea pe tot parcursul vieții.

REFERENCES

- [1] BOROVİK, A., *Mathematics under the microscope: Notes on cognitive aspects of mathematical practice*, American Mathematical Society, 2010.
- [2] CORMEN, T. H., LEISERSON, C. E., RIVEST, R. L., and STEIN, C., *Introduction to Algorithms*, MIT Press, 2009.
- [3] GERSTING, J. L., *Mathematical Structures in Computer Science*, W.H. Freeman and Company, 1990.
- [4] GROVER, S., and PEA, R., *Computational thinking in K-12: A review of the state of the field*, Educational Researcher, **42** (1) (2013), 38–43.
- [5] NISTOR, A., și ISTRATE, O., *Didactica informaticii și noile paradigme educaționale*, Iași: Polirom, 2018.
- [6] PĂUN, E., și POTOLEA, D., *Pedagogie. Fundamentări teoretice și demersuri aplicative*, Iași: Polirom, 2002.
- [7] WING, J. M., *Computational Thinking*, Communications of the ACM, **49** (3) (2006), 33–35.

Colegiul Național "Onisifor Ghibu" Oradea, Catedra de Matematică
Strada Onisifor Ghibu 3, Oradea 410578, România
e-mail: corinacurila@yahoo.com