0.01em
01em

# Problema cifrelor impare in puterile lui 2

Facultatea de Matematica si Informatica

Limbaje Formale si Automate

Grupa: 141

Rachieriu Gheorghe Gabriel

May 2, 2025

# Contents

**1 Descrierea problemei** **2**

**2 Ce este de fapt ridicarea la putere?** **2**

**3 Ce este interesant la cazul particular $2^k$** **2**

**4 Interpretare** **2**

4.1 Cazul particular $c_1, c_2, ...c_x \in \{0, 1, 2, 3, 4\}$ . . . 2

4.2 Cazul particular $c_1, c_2, ..., c_k, ...c_x \in \{0, 1, 2, 3, 4\}$ dar $x_k \in \{5, 6, 7, 8, 9\}$ . . . 2

4.3 Mai multe cazuri particulare. Este ceva special cu 2048? . . . 2

**5 Realizarea unui script care să calculeze şirul** **3**

**6 Concluzie** **3**

# 1 Descrierea problemei

Fie sirul infinit definit prin $x_n = 2^n, \quad n \in \mathbb{N}$ .

Primii termeni ai sirului sunt:

1, pentru n=0
**2, pentru n=1**
**4, pentru n=2**
**8, pentru n=3**
16, pentru n=4
32, pentru n=5
**64, pentru n=6**
128, pentru n=7
256, pentru n=8
512, pentru n=9
1024, pentru n=10
**2048, pentru n=11**

Observam ceva interesant pentru n=2,4,8,11. Toate cifrele numerelor din sir sunt pare. In urma generarii sirului pe calculator se observa ca 2048 (al 11-lea termen) este ultimul din sir care are toate cifrele pare. Interesant insa despre aceasta problema este generalizarea, si demonstratia.

# 2 Ce este de fapt ridicarea la putere?

Fie $n, k \in \mathbb{N}$, spunem ca $n^k$ este notatia inmultirii repetate a lui n, cu el insusi, de k ori. În acest sens, rezultatul calculului $n^k$ este n*n*n...n, de k ori.

# 3 Ce este interesant la cazul particular $2^k$

Conform definiției anterioare, rezultatul calcului $2^k$ este $2 * 2 * \cdots * 2$ de $k$ ori. Înmulțirea, la rândul ei, este o adunare repetată; în caz particular, înmulțirea cu 2 se rezumă la adunarea unui număr cu el însuși. În acest context, problema calculului lui $2^k$ este de fapt studiul șirului $T_n = T_{n-1} - T_{n-1}, \quad n \in \mathbb{N}$.

# 4 Interpretare

## 4.1 Cazul particular $c_1, c_2, ...c_x \in \{0, 1, 2, 3, 4\}$

Considerăm că ne aflăm la pasul $i$, iar $T_{i-1}$ este de forma $\overline{c_1 c_2 \ldots c_x}$, unde $x \in \mathbb{N}$. Atunci $T_i = \overline{2c_1 2c_2 \ldots 2c_x}$, deoarece nu există transport şi suma oricăror 2 cifre este mai mică strict decât 10.

## 4.2 Cazul particular $c_1, c_2, ..., c_k, ...c_x \in \{0, 1, 2, 3, 4\}$ dar $x_k \in \{5, 6, 7, 8, 9\}$

Observãm cã existența a cel puțin unei cifre din mulțimea {5,6,7,8,9} produce un transport care care transformã termenul $T_i$ în $\overline{2c_1 2c_2 ... 2c_i + 1 ... 2c_x}$, ale cãrui cifre nu sunt strict pare. Exemplu: 2048 + 2048 = 40**9**6(Explicație: 4+4=8+1(Transport)=9.
Acest tipar, însã, nu demonstreazã inexistența unui numãr cu toate cifrele pare, deoarece cifrele cãt mai semnificative cresc, în mod evident mai încet decât cele mai puțin semnificative. Exemplu: ...072( $2^{17}$)+ ...072($2^{17}$) = ...144($2^{18}$) + ...144($2^{18}$) = ...4288($2^{19}$)(Deşi cifra 2 a produs transport în cifra 3, după câteva iterații, cifrele 1,2 şi 3 revin toate pare).

## 4.3 Mai multe cazuri particulare. Este ceva special cu 2048?

Primii termeni ai şirului $T_n$ sunt: 1, pentru n=0
**2, pentru n=1**
**4, pentru n=2**
**8, pentru n=3**
16, pentru n=4 **Primul Transport**
32, pentru n=5

**64, pentru n=6**
128, pentru n=7 **Transport**
256, pentru n=8 **Transport**
512, pentru n=9 **Transport**
1024, pentru n=10 **Cifre care nu produc transport**
**2048, pentru n=11**
4096, pentru n=12 **Transport**
8192, pentru n=13 **Transport**
16384, pentru n=14 **Transport**
32768, pentru n=15 **Transport**
65536, pentru n=16 **Transport**
131072, pentru n=17 **Transport**
262144, pentru n=18 **Transport**
524288, pentru n=19 **Transport**
1048576, pentru n=20 **Transport**
2097152, pentru n=21 **Transport**
4194304, pentru n=22 **Transport**
8388608, pentru n=23 **Transport**
16777216, pentru n=24 **Transport**
33554432, pentru n=25 **Transport**

Observãm ca de la un n foarte mare, transporturile apar inevitabil.

## 5 Realizarea unui script care să calculeze şirul

O soluție la prima mână ar fi să adun numerele cu ele însăşi sau sa le înmulțesc cu 2 repetat ca int-uri($2^{31}$) sau long long-uri($2^{63}-1$). O abordare de care mi-am adus aminte din liceu ar fi adunarea numerelor mari ca string-uri, tinand cont de carry. Cu aceasta metoda, as putea calcula $2^{4,611,686,018,427,387,897}$ (4,611,686,018,427,387,897 este dimensiunea maxima a unui string din STL) care este mai mare decat numarul calculat deja de $2^{133477987019}$ dacă nu aş fi limitat de specificațile calculatorului meu, timp si curent. Personal, pana am scris acest paragraf, am rulat script-ul, si pana la $2^{597592}$ nu mai este niciun numar cu toate cifrele pare. Codul sursă pe GitHub.

## 6 Concluzie

Studiul cifrelor impare în puterile lui 2 relevă o proprietate interesantă: până la $2^{11} = 2048$, există termeni care conțin doar cifre pare, dar începând cu $2^{12}$ apar inevitabil cifre impare, datorate fenomenului de transport în operațiile de adunare repetată.

Prin simularea procesului de înmulțire cu 2 ca adunare de sine și implementarea algoritmului cu stringuri pentru a evita limitările tipurilor de date, am putut observa cum, indiferent de strategia inițială, apariția transporturilor devine inevitabilă în evoluția exponențială a lui $2^n$. Astfel, 2048 devine un reper interesant, dar nu unic, în comportamentul acestui șir.

Prin această analiză, observăm cum concepte de bază din teoria numerelor și programare algoritmică pot fi aplicate pentru a înțelege mai bine proprietăți ascunse ale unor operații matematice aparent simple.