## Diferențele între informatică și matematică

1) Elementul neutru $\quad a+b=a \quad b\neq 0 \quad 1+10^{-6}=1$

2) Comutativitatea $\quad f(a)$: return a++

$a=1$: $\quad$ scrie $(a_1 + f(a)_1) = 2$

scrie $(f(a)_1 + a_2) = 3$

3) Asociativitatea

$\underbrace{\underbrace{0,5\times 0,7}_{0,3}\times 0,9}_{0,2}$ trunchie

$\neq$

$\underbrace{0,5\times \underbrace{0,7\times 0,9}_{0,6}}_{0,3}$

Teoremă: În $\mathbb{R}$, $\exists\ \alpha, \beta \notin \mathbb{R}$ cu $\alpha^{\beta}\in\mathbb{Q}$

$x=\sqrt{2}^{\sqrt{2}}$

1) $x\in\mathbb{Q}$ $\quad$ O.K

2) $x\notin\mathbb{Q}$ $\to$ $x^{\sqrt{2}}=\left(\sqrt{2}^{\sqrt{2}}\right)^{\sqrt{2}}=\sqrt{2}^{2}=2$ $\quad$ OK

$\left(\sqrt{3}\right)^{\log_{\sqrt{3}}2}=2$

Nu ne interesează în informatică

---

## Algoritmi

0) Să arătăm că există algoritm care rezolvă problema
1) Descriem algoritmul
2) Corectitudinea lui
3) Timpul de lucru
4) Optimalitate

Corectitudine

1) Corectitudine parțială | Dacă alg. se termină, rez. este cel dorit ⊠
2) Terminare | algoritmul se termină ⊠

⊠ pentru orice date de input

---

$a\cdot b$

Metoda țăranului rus

```
x ← a ; y ← b ; p ← 0
while x > 0                      [xy + p = a·b] invariant
    if x impar then p ← p + y
    x ← x/2 ; y ← y + y
write (p)
```

Comparăm cu 0
împărțim la 2
Suma a 2 numere

Introducem invarianți
↳ instrucțiune / afirmație cu care ajungem la rez.

(*) invariant

- Suma dată: $a \cdot b + p = a \cdot b$

Presupunem adevărat pt. $(x, y, p)$ și trec la următorul $(x', y', p')$
$xy + p = a \cdot b$ $\longrightarrow$ vrem $x'y' + p' = a \cdot b$

Caz 1: $x = 2k+1$ $\quad p' = p + y \rightarrow x' = k \quad y' = 2y$

$x'y' + p' = k \cdot 2y + p + y = p + y(2k+1) = p + x \cdot y = a \cdot b$

Caz 2: $x = 2k$ $\quad p' = p \rightarrow x' = k \quad y' = 2y$

$x'y' + p' = 2ky + p = xy + p = ab$

Verificare

$a = 54 \quad b = 12$

| x | y | p |
|---|---|---|
| 54 | 12 | 0 |
| 27 | 24 | 0 |
| 13 | 48 | 24 |
| 6 | 96 | 72 |
| 3 | 192 | 264 |
| 1 | 384 | $\boxed{648}$ |
| 0 | | |

1) Dacă se termină, se termină când $x = 0$
$\cancel{xy} + p = ab$

2) $(x_n, y_n, p_n)$ $\{x_n\} \searrow$
la un moment dat $x = 0$

Timp de lucru

$n$ = numărul datelor de intrare

$T(n) = n^3 + 3 \log n + 4 n \log n$ $\rightarrow$ Este de ordinul $O(n \log n)$

$a(a_1, \ldots, a_n)$. $\quad m = \min(a_1, \ldots, a_n)$

$m \leftarrow a_1$
for $i = 2, n$
 if $a_i < m$ then $m \leftarrow a_i$

Acest algoritm este optim

$T(n) = n - 1$ comparații

Fie A un algoritm oarecare

$\forall n$, calculul minimului necesită $\geq n-1$ comparații

$n \in a_1$ prin renumerotare $\rightarrow$ renumerotare de $n-1$ elem.

$\Rightarrow$ trebuie $\geq n-2$ comparații

$\hookrightarrow$ $a_1$ comparat

$a = (a_1, \dots, a_n)$

$m = \min(a_1, \dots, a_n)$ $\quad n-1$

$M = \max(a_1, \dots, a_n)$ $\quad \underline{n-1}$

$2n-2$

$(a_1, \dots a_k, a_{k+1}, a_{k+2}, \dots$

$m, M$

if $a_{k+1} \nless a_{k+2}$ then $a_{k+1} ? m$, $a_{k+2} ? M$
else $a_{k+1} ? M$, $a_{k+2} ? m$ $\Big]$ ~~4~~ 3 comparații

- $n = 2k$

$a_1 ? a_2$ $\quad m, M$ $\quad$ 1 comp

De $k-1$ ori fiecare 3 comparații $\quad \frac{3k-3}{3k-2}$

- $n = 2k+1$

$m = M = a_1$

$k$ perechi $3k$ comparații

$T(n) = \left\lceil \frac{3n}{2} \right\rceil - 2 < 2n-2$

¿ IS THIS OPTIMAL?

$n = 2k$ $\quad 3k-2$

$n = 2k+1$ $\quad \left\lceil \frac{6k+3}{2} \right\rceil - 2 \quad 3k + \left\lceil \frac{3}{2} \right\rceil - 2$

$2 - 2$

Orice algoritm care calculează $\begin{cases} m \\ M \end{cases}$ necesită $\geq \left[\frac{3n}{2}\right] - 2$

a) Există algoritm care rezolva problema

→ Există probleme pentru care nu se pot elabora algoritmi

Problema terminării programului nu este decidabilă
(nu există alg. pentru ea)

Problema Echivalentei programului

Intrare: 2 programe

Ieșire: Sunt echivalente sau nu