

Examen CC

Problem 1

$$Y \leftarrow 1 \quad E \leftarrow 1$$

$$X_1 \leftarrow 2 \quad A_1 \leftarrow 2$$

$$Z_1 \leftarrow 3 \quad A_2 \leftarrow 3$$

$$X_2 \leftarrow 4$$

$$Z_2 \leftarrow 5$$

$$X_3 \leftarrow 6$$

$$\text{IF } X_3 \neq 0 \text{ GOTO } A_2 = \langle 0, \langle 3+2, 6 \rangle \rangle = \langle 0, 351 \rangle = 702$$

$$Y \leftarrow Y + 1 = \langle 0, \langle 1, 0 \rangle \rangle - \langle 0, 1 \rangle = 2$$

$$A_1: X_1 \leftarrow X_1 - 1 = \langle 2, \langle 2, 1 \rangle \rangle - \langle 2, 11 \rangle = 91$$

Am folosit formula din curs $\langle a, b \rangle = 2^a(2b+1)-1$.

Vrem să Godelizăm $[702, 2, 91]$: $2^{702} \cdot 3^2 \cdot 5^{91}$

$$\cancel{2^{702}} \cdot \cancel{3^2} \cdot \cancel{5^{91}} = 1$$

Asadar, encodarea instructiunilor este $2^{702} \cdot 3^2 \cdot 5^{91} - 1$.

Calculate:

IF $x_3 \neq 0$ GOTO A_2

Label = 0

variable = $\#(x_3) = 6$

tip = $\#(A_2) + 2 = 3 + 2 = 5$

$$\Rightarrow \langle 0, \langle 5, 6-1 \rangle \rangle = \langle 0, 2^5(2 \cdot 5 + 1) - 1 \rangle = \langle 0, 351 \rangle \\ = 702$$

$y \leftarrow y + 1$

Label 0

variable: 1

tip: 1

$$\Rightarrow \langle 0, \langle 1, 1-1 \rangle \rangle = \langle 0, \langle 0, 0 \rangle \rangle = \langle 0, 1 \rangle = 2$$

$A_1: x_1 \leftarrow x_1 - 1$

Label: $\#(A_1) = 2$

variable: $\#x_1 = 2$

tip: 2

$$\Rightarrow \langle 2, \langle 2, 2-1 \rangle \rangle = \langle 2, \langle 2, 1 \rangle \rangle = \langle 2, 11 \rangle = 91$$

Examen CC

Problema 2

Idee:

Folosim o MT deterministă pe două benzi.
Pe prima bandă citim inputul, fără să scriem
niciodată pe ea, având asadar o MT
off-line.

Pe a doua bandă ținem un contor în baza 2,
pe care îl incrementăm cu câte 1 când ne deplasăm
la un cuvânt mai în dreapta.

ÎE. A doua bandă va conține $\text{bin}(1)$ când îl
verificăm pe w_1 , $\text{bin}(2)$ când îl verificăm pe
 w_2 , ..., $\text{bin}(n)$ când îl verificăm pe w_n .

Pentru comoditate, ținem numărul de pe banda
2 în reverse. De ex. la w_{10} dorim a doua
bandă să arate 1011011B...

Pasii MT:

$B_1: \# w_1 \# w_2 \dots \# w_n$
↑

$B_2: 1 \ 1 \ 0 \ 1 \dots 1$
↑

Pas 1: adăugăm 1

$B_2 = \text{bin}(x) \text{ reversed}$

la banda B_2 , și sărim peste

'#'. Dacă nu avem ~~niciun~~ '#' pe banda 1, atunci inputul nu este corect și refuzăm.

Pas 2: Dacă pe banda B_1 am dat de Blank, atunci refuzăm inputul.

Dacă cuvântul w_i pe care ne aflăm pe B_1 este egal cu numărul de pe B_2 (inversat), atunci acceptăm inputul.

Pas 3: ~~Sărim~~ Ne deplasăm pe B_1 până la '#' sau Blank în dreapta, și sărim la Pas 1, dacă avem '#', sau refuzăm dacă avem Blank.

Explicații suplimentare:

- Creșterea lui B_2 cu 1:

- setăm cursorul la începutul benzii
- cât timp cursorul citește 1, scriem 0 și ne deplasăm la dreapta
- scriem 1.

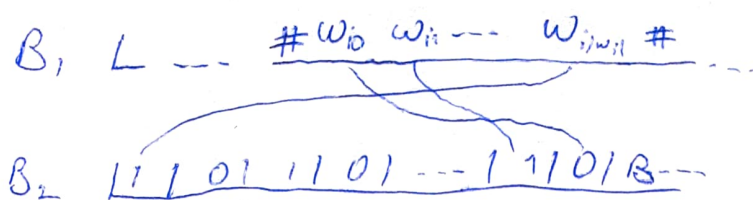
Deși poate face până la $\log(\text{valoare})$ pași, complexitatea amortizată de incrementare de n ori este:

$$\text{Inc} = \underbrace{\frac{1}{2} \cdot 1}_{\substack{\uparrow \\ \text{bit 0 este 0}}} + \underbrace{\frac{1}{4} \cdot 2}_{\substack{\uparrow \\ \text{bit 0 este 1} \\ \text{bit 1 este 0}}} + \dots + \underbrace{\frac{1}{2^n} \cdot n}_{\substack{\uparrow \\ \text{bit 0 este 1} \\ \vdots \\ \text{bit } n-1 \text{ este 1} \\ \text{bit } n \text{ este 0}}} + \dots$$

Prin calcul, $\text{Inc} = 2$, adică este constant.

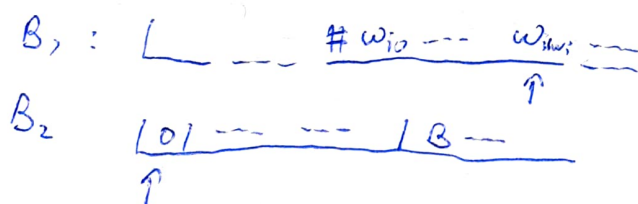
Asadar, incrementarea cu 1 poate fi considerată $O(1)$.

- verificarea că $w_1 = B_2$ reversed



Doim să efectuăm verificarea în $O(|w|)$.

Pentru asta, mutăm acul de citire de la începutul lui w_i la sfârșitul lui w_i . Acul de citire pe B_2 rămâne la începutul benzii.



Verificăm în $|w|$ pasi dacă cele două siruri sunt egale, crescând acul pe B_2 și scăzând pe B_1 , oprindu-ne la prima diferență. Indiferent dacă acceptăm sau nu, readucem acul de pe B_1 la sfârșitul cuvântului, și de pe B_2 la începutul benzii.

Complexitate:

Fie w input ul. Atadar, $|w| \geq 2 \cdot n > n$.

Timp: - incrementarea de pe B_2 se efectuează de n ori, în $O(1)$ dar în total în $O(n)$
- parcurgerea și verificarea lui w se efectuează prin 2 parcurgeri ale fiecărui cuvânt, dar în timp liniar în $|w|$.

Atadar, complexitatea finală este

$O(|w| + n) = O(|w|)$, adică timp liniar în lungimea inputului.

Spatiu: - MT este off-line.

- pe B_2 vom avea maxim $\lceil \log_2(n) \rceil$ elemente.

Atadar, complexitatea spațiu este

$O(\log(n)) \subseteq O(\log(|w|))$

Avem asadar $O(\text{input})$ timp (polinomial)

$O(\log(\text{input}))$ spațiu (sub-polinomial)

De asemenea, observăm că MT nu poate
cădea la infinit (se oprește pe fiecare intrare).