

17. Concepeți și implementați un algoritm care, folosind o stivă, efectuează transformarea unui număr din baza 10 într-o bază de numerație dată ca parametru.

$X_{(10)} \rightarrow ?_{(k)}$

Algoritmul de conversie prin împărțiri successive

Toate calculele se fac în baza de sursă (10).

Se împarte nr. (X) la baza destinație (k) => cât și un rest

Se ia câtul, se împarte la baza destinație => cât și un rest

Se ia câtul, se împarte la baza destinație => cât și un rest

...

Până când câtul = 0.

Nr. în baza k se obține luând resturile în ordine inversă.

Vom folosi stiva pt. a păstra resturile

Subalgoritm Conversie (x , y, k)

//precondiții: x este un nr. în baza 10, k este noua bază de numerație, $k \in [2,16]$

//postcondiții: $y_{(k)} = x_{(10)}$, y este un șir de caractere care reprezintă nr. în baza k

s ← create() //se creează o stivă O(1)

Cât timp $x \neq 0$ execută //n ori

 push(x mod k,s) //am salvat restul O(1)

$x \leftarrow x \text{ div } k$ //x devine câtul

sf_cât timp

//construim nr. y

y ← ""

Cât timp ¬isEmpty(s) execută //n ori (isEmpty O(1))

 c ← pop(s) // O(1)

 Dacă $c < 10$ atunci

 @Concatenează(y,c+'0')

 Sf_Dacă

 Dacă $c = 10$ atunci

 @Concatenează(y,'A'+c-10)

 Sf_Dacă

Sf_Cât timp

Dacă $y = ""$ atunci $y \leftarrow "0"$ sf_Dacă

Sf_Subalgoritm

$(x = y_n y_{n-1} \dots y_1 y_0)_{(k)} = y_n * k^n + y_{n-1} * k^{n-1} + \dots + y_1 * k^1 + y_0 * k^0 \approx k^n$

n – nr. de cifre în baza k , $x \approx k^n \Rightarrow n \approx \log_k x \Rightarrow n \approx \log_2 x$ (adică n e aprox. $\log x$) \Rightarrow complexitatea subalg. Conversie este $\log_2 x$ (caz favorabil, mediu și cel mai rău caz)

Folosind o coadă, să se convertească un nr. y din baza k în baza 10.

Ideea de rezolvare:

Vom crea din șirul de litere/cifre y o coadă cu cifrele nr. y începând cu cea mai semnificativă

$$x_{(10)} = y_n y_{n-1} \dots y_1 y_0^{(k)} = y_n * k^n + y_{n-1} * k^{n-1} + \dots + y_1 * k^1 + y_0 * k^0 = ((y_n * k + y_{n-1}) * k + \dots + y_1) * k + y_0$$

Subalgoritm Conversie_in_10 (y, k, x)

//precondiții: y este un șir de caractere care reprezintă nr. în baza k , $k \in [2, 16]$

//postcondiții: x este nr. în baza 10, $x_{(10)} = y_{(k)}$

 QUEUEinit(q)

 //creăm coada cu cifre

 Pentru $i \leftarrow \overline{1, @lungimea(y)}$ execută

 Dacă ' 0 ' $\leq y_i \wedge y_i \leq '9'$ atunci

 QUEUEput($q, y_i - '0'$)

 Altfel

 QUEUEput($q, y_i - 'A' + 10$)

 Sf_Dacă

 Sf_Pentru

$x \leftarrow$ QUEUEget(q)

 Cât timp \neg QUEUEempty(q) execută

$x \leftarrow x * k +$ QUEUEget(q)

 Sf_Cât timp

Sf_Subalgoritm

Complexitatea: $O(n)$ unde n este nr. de caractere al lui $y =$ nr. de cifre al nr. din baza k . (caz favorabil, mediu și cel mai rău caz)

TEMĂ : Termen de predare 1.05.2020

Trebuie să compuneți 1 problemă care se rezolvă cu stive și una cu cozi, să le implementați și să evaluați complexitatea algoritmilor implementați.