

7. táblás gyakorlat feladatai

1. Egy nem üres szekvenciális inputfájlban egy bank ügyfeleinek havi kivét/betét forgalmát (tranzakcióit) tárolják. Egy tranzakció az ügyfél számlaszámából, a tranzakció dátumából, és összegéből (előjeles egész szám: negatív a kivét, pozitív a betét) áll. A tranzakciók számlaszám szerint rendezetten helyezkednek el. Gyűjtsük ki azon számlaszámokat, ahol volt 100000 Ft-ot meghaladó kivétel, és adjuk meg azt a számlaszámot is, amelynél a legkisebb volt a havi forgalom (kivételek és betétek együttes darabszáma).

Specifikáció:

$$A = (x:\text{infile}(\text{Tranz}), y:\text{outfile}(\mathbb{S}), sz:\mathbb{S}) \quad \text{Tranz} = \text{rec}(\text{száml}:\mathbb{S}, \text{dátum}:\mathbb{S}, \text{össz}:\mathbb{Z})$$

$$Ef = (x = x_0 \wedge x \nearrow \text{száml} \wedge |x| > 0)$$

(az $x \nearrow \text{száml}$ azt jelzi, hogy x számlaszám szerint növekedően rendezett)

Ötlet:

Soroljuk fel számlaszámonként a tranzakciók összesített egyenlegeit, hogy ezekből kiválogathassuk a -100000-nél kisebb egyenlegűeket, valamint adjuk meg, hány tranzakció történt az adott számlán.

Új specifikáció:

$$A = (t:\text{enor}(\text{Egyenleg}), y:\text{outfile}(\mathbb{S}), sz:\mathbb{S}) \quad \text{Egyenleg} = \text{rec}(\text{száml}:\mathbb{S}, \text{egy}:\mathbb{Z}, \text{db}:\mathbb{N})$$

$$Ef = (t = t_0 \wedge |t| > 0)$$

$$Uf = (y = \bigoplus_{e \in t_0} \langle e.\text{száml} \rangle \wedge (\min, \text{elem}) = \mathbf{MIN}_{e \in t_0} e.\text{db} \wedge sz = \text{elem}.\text{száml})$$

e.egy < -100000

Összegzés (kiválogatás)

$$t:\text{enor}(E) \sim \text{tenor}(\text{Egyenleg})$$

$$f(e) \sim \langle e.\text{száml} \rangle \text{ ha } e.\text{egy} < -100000$$

$$s \sim y$$

$$H, +, 0 \sim \mathbb{S}^*, \oplus, <$$

Maximum kiválasztás (minimum)

$$t:\text{enor}(E) \sim \text{tenor}(\text{Egyenleg})$$

$$f(e) \sim e.\text{db}$$

$$\max \sim \min$$

$$H, > \sim \mathbb{Z}, <$$

A két tételt közös felsorolás felett végezzük. De a minimum kiválasztást itt nem lehet összegzésként felírni (az \mathbb{N} halmaznak nincs maximuma), ezért a kiválogatás első lépését a minimum kiválasztáshoz hasonlóan a ciklus előtt végezzük el. (Az elem változó használata helyett elég az sz változóval dolgozni.)

Algoritmus:

t.first()	
t.current().egy < -100000	
y : write(t.current().száml)	-
min, sz := t.current().db, t.current().száml	
t.next()	
¬t.end()	
t.current().egy < -100000	
y : write(t.current().száml)	-
t.current().db < min	
min, sz := t.current().db, t.current().száml	-
t.next()	

Felsoroló:

t:enor(Egyenleg) Egyenleg = rec(száml:S, egy:ℤ, db:ℕ)

Egyenleg*	first()	next()	current() : Egyenleg	end() : ℤ
x : infile(Tranz) dx : Tranz sx : Status akt : Egyenleg vége : ℤ	sx,dx,x:read next()	lásd külön	return akt	return vége

Tranz = rec(száml:S, dátum:S, össz:ℤ)

next() művelet

A = (x:infile(Tranz), dx:Tranz, sx:Status, akt:Egyenleg, vége:ℤ)

Ef = (x = x' ∧ x.száml = dx.száml ∧ dx = dx' ∧ sx = sx')

Uf = (vége=(sx'=abnorm) ∧ (¬vége → akt.száml=dx'.száml ∧
 $\wedge ((\text{akt.egy}, \text{akt.db}), (sx, dx, x)) = \sum_{dx \in (dx', x')} dx.száml = akt.száml (dx.össz, 1))$))

Dupla összegzés

t:enor(E) ~ x:infile(Tranz)
 (sx,dx,x:read) first() nélkül,
 amíg: dx.száml=akt.száml
 f(e) ~ (dx.össz, 1)
 s ~ (akt.egy, akt.db)
 H, +, 0 ~ (ℤ×ℕ), (+, +), (0, 0)

vége := sx=abnorm	
¬vége	
akt := (dx.száml, 0, 0)	-
sx=norm ∧ dx.száml=akt.száml	
akt.egy := akt.egy + dx.össz akt.db := akt.db + 1	
sx,dx,x:read	

2. Egy szekvenciális inputfájl vadászok eredményeit (vadász neve, vadászat dátuma, zsákmány fajtája, zsákmány súlya) tárolja a vadászok neve, azon belül a vadászat dátuma szerint rendezetten.

a) Igaz-e, hogy minden vadász valamelyik vadászatán lőtt medvét?

Specifikáció:

$$A = (x:\text{infile}(\text{Trófea}), l:\mathbb{L})$$

$$\text{Trófea} = \text{rec}(\text{név}:\mathbb{S}, \text{dátum}:\mathbb{S}, \text{fajta}:\mathbb{S}, \text{súly}:\mathbb{N})$$

$$Ef = (x=x_0 \wedge x \nearrow (\text{név}, \text{dátum}))$$

Ötlet:

Soroljunk fel annyi logikai értéket ahány vadász van: egy logikai érték akkor igaz, ha a vadász lőtt medvét.

Új specifikáció:

$$A = (t:\text{enor}(\mathbb{L}), l:\mathbb{L})$$

$$Ef = (t=t_0)$$

$$Uf = (l = \forall \text{SEARCH}_{e \in t_0} e)$$

Optimista lineáris keresés

$$t:\text{enor}(E) \sim t:\text{enor}(\mathbb{L})$$

$$\text{felt}(e) \sim e$$

Algoritmus:

$l := \text{igaz}$ $t.\text{first}()$
$l \wedge \neg t.\text{end}()$
$l := t.\text{current}()$
$t.\text{next}()$

Felsoroló: $t:\text{enor}(\mathbb{L})$

\mathbb{L}^*	first()	next()	current() : \mathbb{L}	end() : \mathbb{L}
$x : \text{infile}(\text{Trófea})$ $dx : \text{Trófea}$ $sx : \text{Status}$ $\text{akt} : \mathbb{L}$ $\text{vége} : \mathbb{L}$	$sx, dx, x:\text{read}$ $\text{next}()$	lásd külön	return akt	return vége

$$\text{Trófea} = \text{rec}(\text{név}:\mathbb{S}, \text{dátum}:\mathbb{S}, \text{fajta}:\mathbb{S}, \text{súly}:\mathbb{N})$$

next() művelet (Lőtt-e medvét a soron következő vadász?)

$$A = (x:\text{infile}(\text{Trófea}), dx:\text{Trófea}, sx:\text{Status}, \text{akt}:\mathbb{L}, \text{vége}:\mathbb{L})$$

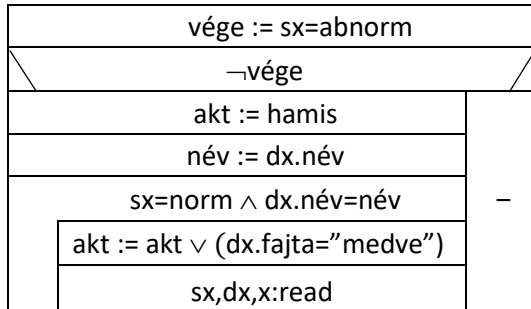
$$Ef = (x = x' \wedge x \nearrow \text{név} \wedge dx = dx' \wedge sx = sx')$$

$$Uf = (\text{vége} = (sx' = \text{abnorm}) \wedge (\neg \text{vége} \rightarrow (\text{akt}, (sx, dx, x)) = \bigvee_{dx \in (dx', x')} dx.\text{név} = dx'.\text{név} \wedge dx.\text{fajta} = \text{medve})))$$

Az eldöntéshez a lineáris keresés nem elég, mert végig kell menni egy vadász összes trófeáján.

Összegzés (összevagyolás)

t:enor(E) ~ x:infile(Trófea) (sx,dx,x:read)
 first() nélkül,
 amíg: dx.név=dx'.név
 f(e) ~ dx.fajta="medve"
 s ~ akt
 H, +, 0 ~ \mathbb{L} , \vee , hamis



b) A fenti mellett mondjuk meg azt is, hogy hány olyan vadász van, hogy zsákmányai összszúlya meghaladja a 250 kilogrammot.

Specifikáció:

A = (x:infile(Trófea), l: \mathbb{L} , c: \mathbb{N})
 Trófea = rec(név:S, dátum:S, fajta:S, súly:N)
 Ef = (x=x₀ \wedge x[↑](név,dátum)

Ötlet:

Soroljuk fel, hogy az egyes vadászok lóttak-e medvét, és mennyi a zsákmányaiknak az összszúlya.

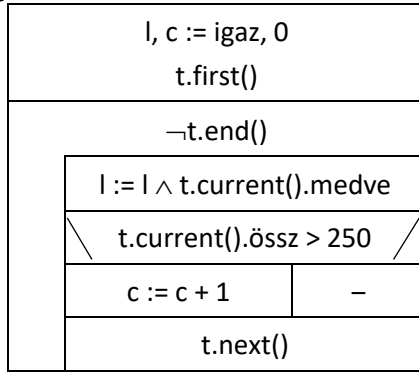
Új specifikáció:

A = (t:enor(Vadász), l: \mathbb{L} , c: \mathbb{N})
 Vadász = rec(medve: \mathbb{L} , össz: \mathbb{N})
 Ef = (t=t₀)
 Uf = (l = $\bigwedge_{e \in t_0} e.medve \wedge c = \sum_{\substack{e \in t_0 \\ e.össz > 250}} 1$)

Összeéselés és számlálás közös felsorolón

t:enor(E) ~ t:enor(Vadász)
 f(e) ~ e.medve
 H, +, 0 ~ \mathbb{L} , \wedge , igaz
 felt(e) ~ e.össz>250

Algoritmus:



Felsoroló:

t:enor(Vadász)		Vadász = rec(medve:ℒ, össz:ℕ)		
Vadász*	first()	next()	current() : Vadász	end() : ℒ
x : infile(Trófea) dx : Trófea sx : Status akt : Vadász vége : ℒ	sx,dx,x:read next()	lásd külön	return akt	return vége

Trófea = rec(név:ℒ, dátum:ℒ, fajta:ℒ, súly:ℕ)

next() művelet

$A = (x : \text{infile}(\text{Trófea}), \text{akt}:\text{Vadász}, \text{vége}:\mathbb{L})$

$Ef = (x = x' \wedge dx = dx' \wedge sx = sx' \wedge x \nearrow_{(\text{név}, \text{dátum})})$

$Uf = (\text{vége} = (sx = \text{abnorm}) \wedge (\neg \text{vége} \rightarrow$

$(\text{akt.medve}, (sx, dx, x)) = \bigvee_{dx \in (dx', x')}^{dx.név = dx'.név} dx.fajta = \text{"medve"})$

$\wedge (\text{akt.össz}, (sx, dx, x)) = \sum_{dx \in (dx', x')}^{dx.név = dx'.név} dx.súly)$

Két összegzés közös felsorolóval

t:enor(E) ~ x:infile(Trófea)

(sx,dx,x:read) first() nélkül,

amíg: dx.név=dx'.név

f(e) ~ (dx.fajta="medve", dx.súly)

s ~ akt

H, +, 0 ~ (ℒ×ℕ), (∨, +), (hamis, 0)

