

6. táblás gyakorlat feladatai

- Gyűjtsük ki egy szekvenciális inputfájlban rendezve tárolt egész számok közül azt, hogy melyik számból hány darab található.

Specifikáció:

$A = (x:\text{infile}(\mathbb{Z}), y:\text{outfile}(\text{Össz}))$
 $\text{Össz} = \text{rec}(\text{szám}:\mathbb{Z}, \text{db}:\mathbb{N})$
 $Ef = (x = x_0 \wedge x \nearrow)$
 ($x \nearrow$ azt jelzi, hogy az x növekedően rendezett)

Új Specifikáció:

$A = (t:\text{enor}(\text{Össz}), y:\text{outfile}(\text{Össz}))$
 $Ef = (t = t_0)$
 $Uf = (y = \bigoplus_{e \in t_0} \langle e \rangle)$

Összegzés (másolás)

$f(e) \sim \langle e \rangle$
 $s \sim y$
 $H, +, 0 \sim \text{Össz}^*, \oplus, \langle \rangle$

Ötlet:

Soroljuk fel az eredménybe számt rekordokat, és másoljuk output fájlba őket.

Algoritmus:

$y := \langle \rangle$ $t.\text{first}()$
$\neg t.\text{end}()$
$y : \text{write}(t.\text{current}())$
$t.\text{next}()$

Felsoroló:

$t:\text{enor}(\text{Össz}) \quad \text{Össz} = \text{rec}(\text{num}:\mathbb{Z}, \text{count}:\mathbb{N})$

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

next() művelet

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

$Ef = (x = x' \wedge x \nearrow \wedge dx = dx' \wedge sx = sx')$

$dx = \text{akt.num}$

$Uf = (vége = (sx' = \text{abnorm}) \wedge (\neg vége \rightarrow \text{akt.szám} = dx' \wedge (\text{akt.db}, (sx, dx, x)) = \sum_{dx \in (dx', x')} 1))$

Megj: Az összegzésnek két eredménye van: a darabszám (akt.db); és a felsoroló aktuális állapota, amelyet az sx,dx,x változók értékei írnak le a next() művelet végén.

Összegzés (megszámolás)

$t:\text{enor}(E) \sim x:\text{infile}(\mathbb{Z}) (sx, dx, x:\text{read})$
 $\text{first() nélkül, felt: } dx = \text{akt.szám}$
 $f(e) \sim 1$
 $s \sim \text{akt.db}$
 $H, +, 0 \sim \mathbb{N}, +, 0$

$vége := sx = \text{abnorm}$
$\neg vége$
$\text{akt.szám}, \text{akt.db} := dx, 0$
$sx = \text{norm} \wedge dx = \text{akt.szám}$
$\text{akt.db} := \text{akt.db} + 1$
$sx, dx, x:\text{read}$

2. Egy szekvenciális inputfájlban egy banknál számlát nyitott ügyfelek e havi kivét/betét forgalmát (tranzakcióit) tároljuk. Egy tranzakció tartalmazza az ügyfél számlaszámát, a tranzakció dátumát, és az összegét (előjeles egész szám: negatív a kivét, pozitív a betét). A tranzakciók a szekvenciális fájlban számlaszám szerint rendezetten helyezkednek el. Gyűjtjük ki azon számlaszámokat, és az ahhoz tartozó tranzakcióknak az egyenlegét, ahol ez az egyenleg kisebb –100000 Ft-nál!

Specifikáció:

$A = (x:infile(\text{Ügyfél}), y:outfile(\text{Egyenleg}))$
 $\text{Ügyfél} = \text{rec}(\text{száml}:\mathbb{S}, \text{dátum}:\mathbb{S}, \text{össz}:\mathbb{Z})$
 $\text{Egyenleg} = \text{rec}(\text{száml}:\mathbb{S}, \text{egy}:\mathbb{Z})$
 $Ef = (x = x_0 \wedge x \nearrow_{\text{száml}})$

(az $x \nearrow_{\text{száml}}$ azt jelzi, hogy x számla-szám szerint növekedően rendezett)

Új specifikáció:

$A = (t:enor(\text{Egyenleg}), y:outfile(\text{Egyenleg}))$
 $Ef = (t = t_0)$
 $Uf = (y = \bigoplus_{e \in t_0} \langle e \rangle)$
 $e.\text{egy} < -100000$

Összegzés (kiválogatás)

$f(e) \sim \langle e \rangle \text{ ha } e.\text{egy} < -100000$
 $s \sim y$
 $H, +, 0 \sim \text{Egyenleg}^*, \bigoplus, \langle \rangle$

Ötlet:

Soroljuk fel számlaszámonként a tranzakciók összesített egyenlegeit, hogy ezekből kiválogathassuk a -100000-nél kisebb egyenlegűeket.

Algoritmus:

$y := \langle \rangle$	
$t.\text{first}()$	
$\neg t.\text{end}()$	
$t.\text{current}().\text{egy} < -100000$	
$y : \text{write}(t.\text{current}())$	–
$t.\text{next}()$	

Felsoroló:

$t:enor(\text{Egyenleg})$ $\text{Egyenleg} = \text{rec}(\text{száml}:\mathbb{S}, \text{egy}:\mathbb{Z})$

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

$\text{Ügyfél} = \text{rec}(\text{száml}:\mathbb{S}, \text{dátum}:\mathbb{S}, \text{össz}:\mathbb{Z})$

next() művelet

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

$Ef = (x = x' \wedge x \nearrow_{\text{száml}} \wedge dx = dx' \wedge sx = sx')$

$Uf = (vége = (sx' = \text{abnorm}) \wedge (\neg vége \rightarrow akt.\text{száml} = dx'.\text{száml} \wedge (akt.\text{egy}, (sx, dx, x)) = \sum_{dx \in (dx', x')} dx.\text{össz}))$
 $dx.\text{száml} = akt.\text{száml}$

Összegzés

$t:enor(E) \sim x:infile(\text{Ügyfél}) (sx, dx, x:read)$
 $first()$ nélkül,
 $felt: dx.\text{száml} = akt.\text{száml}$
 $f(e) \sim dx.\text{össz}$
 $s \sim akt.\text{egy}$
 $H, +, 0 \sim \mathbb{Z}, +, 0$

vége := sx=abnorm	
¬vége	
akt.száml, akt.egy := dx.száml, 0	–
sx=norm ∧ dx.száml=akt.száml	
akt.egy := akt.egy + dx.össz	
sx,dx,x:read	

3. Számoljuk meg egy karakterekből álló szekvenciális inputfájlban a szavakat úgy, hogy a 12 betűnél hosszabb szavakat duplán vesszük figyelembe! (Egy szót szóközök vagy a fájl vége határol.)

Specifikáció:

$A = (x:\text{infile}(\mathbb{K}), c:\mathbb{N})$
 $Ef = (x = x_0)$

Új specifikáció:

$A = (t:\text{enor}(\mathbb{N}), c:\mathbb{N})$
 $Ef = (t = t_0)$
 $Uf = (c = \sum_{e \in t_0} \{2, \text{ha } e > 12; 1, \text{ha } e \leq 12\})$

Összegzés

$f(e) \sim \{2, \text{ha } e > 12; 1, \text{ha } e \leq 12\}$
 $s \sim c$
 $H, +, 0 \sim \mathbb{N}, +, 0$

Ötlet:

Soroljuk fel a szavak hosszait, és ilyen módon az eredményt már könnyen kiszámolhatjuk.

Algoritmus:

c := 0	
t.first()	
¬t.end()	
t.current() > 12	
c := c + 2	c := c + 1
t.next()	

Felsoroló:

t:enor(\mathbb{N})

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

next() művelet

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

$Ef = (x = x' \wedge dx = dx' \wedge sx = sx')$

$Uf = ((dx'', (sx'', dx'', x'')) = \text{SELECT}_{dx \in (dx', x')} (sx = \text{abnorm} \vee dx \neq ' ') \wedge_{dx \neq ' '}$
 $\text{vége} = (sx'' = \text{abnorm}) \wedge (\neg \text{vége} \rightarrow (akt, (sx, dx, x)) = \sum_{dx \in (dx'', x'')} 1))$

Megj: A kiválasztásnak két eredménye van: a keresett szóköz (dx''), amelyhez nem vezetünk be külön output változót; és a felsoroló aktuális állapota, amelyet az (sx'', dx'', x'') hármas ír le. (Nem okozna félreértést, ha csak az (sx'', dx'', x'') -t tüntetnénk fel a kiválasztás eredményeként.)

Kiválasztás

t:enor(\mathbb{E}) \sim x:infile(\mathbb{K}) (sx,dx,x:read)
first() nélkül

felt(e) \sim sx=abnorm \vee dx \neq ' '

Összegzés (megszámolás)

t:enor(\mathbb{E}) \sim x:infile(\mathbb{K}) (sx,dx,x:read)
first() nélkül, felt: dx \neq ' '

$f(e) \sim 1$
 $s \sim akt$
 $H, +, 0 \sim \mathbb{N}, +, 0$

sx=norm \wedge dx=' '	
sx,dx,x:read	
vége := sx=abnorm	
¬vége	
akt := 0	-
sx=norm \wedge dx \neq ' '	
akt := akt+1	
sx,dx,x:read	

4. Másoljuk át karakterenként egy szekvenciális inputfájl szövegét egy szekvenciális outputfájlba úgy, hogy a szavak között csak egyetlen szóközt tartunk meg!

Specifikáció:

$A = (x:\text{infile}(\mathbb{K}), y:\text{infile}(\mathbb{K}))$
 $Ef = (x = x_0)$

Új specifikáció:

$A = (t:\text{enor}(\mathbb{S}), y:\text{infile}(\mathbb{K}))$
 $Ef = (t = t_0)$
 $Uf = (y = \bigoplus_{e \in t_0} (e \oplus \langle ' \rangle))$

Összegzés (összefűzés)

$f(e) \sim e \oplus \langle ' \rangle$
 $s \sim y$
 $H, +, 0 \sim \mathbb{K}^*, \oplus, \langle \rangle$

Felsoroló

$t:\text{enor}(\mathbb{S})$

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

szóközt_átlép() művelet

$A = (x:\text{infile}(\mathbb{K}), dx:\mathbb{K}, sx:\text{Status})$
 $Ef = (x = x' \wedge dx = dx' \wedge sx = sx')$
 $Uf = ((sx, dx, x) =$
 $\text{SELECT}_{dx \in (dx', x')} (sx = \text{abnorm} \vee dx \neq ' '))$

Kiválasztás

$t:\text{enor}(\mathbb{E}) \sim x:\text{infile}(\mathbb{K}) (sx, dx, x:\text{read})$
 first nélkül
 $\text{felt}(e) \sim sx = \text{abnorm} \vee dx \neq ' '$

$sx = \text{norm} \wedge dx = ' '$
$sx, dx, x:\text{read}$

next() művelet

$A = (x:\text{infile}(\mathbb{K}), dx:\mathbb{K}, sx:\text{Status}, akt:\mathbb{S}, vége:\mathbb{L})$
 $Ef = (x = x' \wedge dx = dx' \wedge sx = sx' \wedge (sx = \text{norm} \rightarrow dx \neq ' '))_{dx \neq ' '}$
 $Uf = (vége = (sx' = \text{abnorm}) \wedge (\neg vége \rightarrow (akt, (sx'', dx'', x'')) = \bigoplus_{dx \in (dx', x')} \langle dx \rangle$
 $\wedge (sx, dx, x) = \text{SELECT}_{dx \in (dx', x')} (sx = \text{abnorm} \vee dx \neq ' ')))$

Összegzés (összefűzés)

$t:\text{enor}(\mathbb{E}) \sim x:\text{infile}(\mathbb{K}) (sx, dx, x:\text{read})$
 first() nélkül, felt: $dx \neq ' '$
 $f(e) \sim \langle dx \rangle$
 $s \sim akt$
 $H, +, 0 \sim \mathbb{K}^*, \oplus, \langle \rangle$

$vége := sx = \text{abnorm}$	
$\neg vége$	
$akt := \langle \rangle$	
$sx = \text{norm} \wedge dx \neq ' '$	–
$akt := akt \oplus \langle dx \rangle$	
$sx, dx, x:\text{read}$	
szóközt_átlép()	

Házi feladat:

Egy étteremben a pincérek által felvett rendeléseket egy szöveges állományban tartják nyilván az ételek neve, azon belül a rendelések időpontja szerint rendezett formában. Feltehetjük, hogy a fájl nem üres. Egy rendelés az asztal sorszámból, a rendelt étel nevéből (sztring), a rendelés időpontjából (sztring), rendelt adagok számából (természetes szám), egy adag árából (természetes szám) áll. Melyik étel hozta az étteremnek a legtöbb bevételt (összesített darab*egységár)?