

Gregorics Tibor
EHACODE.ELTE
gt@inf.elte.hu
0.csoport

2. beadandó/0.feladat

2012. január 11.

Feladat

Az Országos Horgász Bajnokság mindegyik versenyén feljegyezték a versenyzők eredményeit, és egy szöveges állományban rögzítették az adatokat. Ennek minden sora tartalmazza a versenyző horgász és a verseny azonosítóját (szóközök nélküli sztring), majd halfajta-méret párok (szóközök nélküli sztring és természetes szám párok) formájában az adott versenyen elért fogásokat. Az adatok szóközökkel vagy tabulátorjelekkel vannak egy soron belül elválasztva. A szöveges állomány sorait a horgászok szerint rendezték. Feltehetjük, hogy a szöveges állomány helyesen van kitöltve. Példa az állomány egy sorára:

JANIBÁ Kiliti0512 Ponty 45 Ponty 53 Harcsa 96

Nevezzünk meg olyan horgászt, aki minden versenyént legalább két harcsát fogott?

Főprogram terve

$A = (t : \text{Enor}(\text{Horgász}), l : \mathbb{L}, \text{azon} : \text{String}) \quad \text{Horgász} = \text{rec}(\text{azon}:\text{String}, \text{ügyes}:\mathbb{L})$
 $Ef = (t = t')$
 $Uf = (l, \text{azon} = \text{SEARCH}_{e \in t'}(e.\text{ügyes}))$

$l := \text{hamis}; t.\text{first}()$
$\neg l \wedge \neg t.\text{end}()$
$l, \text{azon} := t.\text{current}().\text{ügyes}, t.\text{current}().\text{azon}$
$t.\text{next}()$

Horgászok felsorolója

$\text{enor}(\text{Horgász})$	$\text{first}(), \text{next}(), \text{current}(), \text{end}()$
$tt : \text{enor}(\text{Verseny})$	$\text{first}() \sim tt.\text{first}(); tt.\text{next}()$
$\text{akt} : \text{Horgász}$	$\text{next}() \sim \text{ld. külön}$
$\text{vége} : \mathbb{L}$	$\text{current}() \sim \text{akt}$
	$\text{end}() \sim \text{vége}$

Az $\text{Enor}(\text{Horgász})$ $\text{next}()$ műveletnek az alábbi feladatot kell megoldania:

Állítsuk elő a soron következő horgászt, akiről el kell dönteni, hogy vajon minden versenyén fogott-e legalább két harcsát. Ehhez fel tudjuk sorolni a horgászok versenyeinek eredményét (melyik horgász melyik versenyen hány harcsát fogott, azaz $\text{Verseny} = \text{rec}(\text{horgász}:\text{String}, \text{verseny}:\text{String}, \text{db}:\mathbb{N})$). Az adott horgász legelső versenyét, a $tt.\text{current}()$ értékét már

ismerjük, ehhez nem kell sem a $tt.first()$, sem $tt.next()$. A felsorolás csak addig tart, amíg ugyanazon horgász versenyeit „olvassuk” a $tt.next()$ művelettel.

$$A^{next} = (tt:enor(Verseny), vége:\mathbb{L}, akt:Horgász)$$

$$E^{f^{next}} = (tt = tt^1)$$

$$U^{f^{next}} = (vége = tt.end() \wedge \neg vége \rightarrow akt.ügyes = \bigwedge_{e \in tt'} (e.db \geq 2))$$

next()

vége:= tt.end()	
$\neg vége$	
$akt.azon:=tt.current().horgász$ $akt.ügyes:=tt.current().db \geq 2$	SKIP
$\neg tt.end() \wedge tt.current().horgász=akt.azon$	
$akt.ügyes:= akt.ügyes \wedge tt.current().db \geq 2$	
$tt.next()$	

Versenyek felsorolója

$enor(Verseny)$	$first(), next(), current(), end()$
$f : infile(Sor)$ $akt : Verseny$ $vége : \mathbb{L}$	$first() \sim tld. külön$ $next() \sim ld. külön$ $current() \sim akt$ $end() \sim vége$

Az $Enor(Verseny)$ $first()$ és $next()$ műveletei megegyeznek, és az alábbi feladatot kell megoldaniuk: Olvassuk be a szöveges állomány (az f szekvenciális inputfájl) soron következő sorát. Ha ilyen nincs, akkor a $vége$ változó értéke legyen igaz. Ha van, akkor szedjük össze a beolvasott sorból a horgász és a verseny azonosítóját, majd számoljuk meg a „harcsa” szavakat.

$$A^{next} = (f: infile(Sor), vége:\mathbb{L}, akt:Verseny)$$

$$Sor = seq(Szó)$$

$$E^{f^{next}} = (f = f^1)$$

$$U^{f^{next}} = (sf, df, f = read(f^1) \wedge vége=(sf=abnorm)$$

$$\wedge \neg vége \rightarrow akt.horgász = „df első szava” \wedge akt.verseny = „df második szava”$$

$$\wedge akt.db = „df 'harcsa' szavainak száma”)$$

Az implementációban a fenti két felsoroló objektum (t és tt) osztályait külön fordítási egységbe helyezzük.

Tesztelési terv

A megoldásban három programozási tételt alkalmaztunk: lineáris keresés, optimista lineáris keresés, számlálás.

A. Ügyes horgász keresése (lineáris keresés) tesztelése:

intervallum hossza szerint:

1. Üres állomány.
2. Egyetlen horgász.
3. Több horgász.

intervallum eleje és vége szerint:

4. Első horgász az ügyes.
5. Csak az utolsó horgász ügyes.

tétel szerint:

1. Van ügyes horgász.
2. Nincs ügyes horgász.
3. Több ügyes horgász is van.

B. Horgász ügyességének eldöntése (optimista lineáris keresés) tesztelése:

intervallum hossza szerint:

1. Nincs fogás.
2. Egy horgász egyetlen versenye.
3. Egyetlen horgász több versenye.

intervallum eleje és vége szerint:

4. Egy horgász első versenyén nem fogott két harcsát, a többin igen.
5. Egy horgász utolsó versenyén nem fogott két harcsát, a többin igen.

tétel szerint:

6. Van ügyes horgász.
7. Nincs ügyes horgász.
8. Több ügyes horgász is van.

C. Egy horgász egy versenyen fogott harcsáinak száma (számlálás) tesztelése:

intervallum hossza szerint:

1. Olyan sor, amely nem tartalmaz fogásokat.
2. Olyan sor, amely egy fogást tartalmaz.
3. Olyan sor, amely több fogást is tartalmaz.

intervallum eleje és vége szerint:

4. Olyan sor, amely legelső fogása harcsa.
5. Olyan sor, amely utolsó fogása harcsa.

tétel szerint:

7. Olyan sor, amely nem tartalmaz harcsa fogásokat.
8. Olyan sor, amely egy harcsa fogást tartalmaz.
9. Olyan sor, amely több harcsa fogást tartalmaz.