

OEP - 2. beadandó feladat

Csombordi Viktor - ISG63R, 16-os csoport

2021. április 04

1 Feladat (17)

Egy országos középiskolai sportversenyen feljegyezték a versenyzők eredményeit, és egy szöveges állományban rögzítették az adatokat. A fájl egy sorának felépítése: elsőként a versenyző neve szerepel (két vagy több szóközök nélküli sztring), majd az iskola azonosítója (szóközök nélküli sztring, amely az irányítószámmal kezdődik), majd a versenyző helyezései: sportág-helyezés (szóköz nélküli sztring, pozitív természetes szám) formájában. Az adatok szóközökkel vagy tabulátorjelekkel vannak egy soron belül elválasztva. A szöveges állomány sorait iskola azonosító szerint rendezték. Feltehetjük, hogy a szöveges állomány helyesen van kitöltve. Példa az állomány egy sorára:

Nagyon Ügyes Péter 1063Szinyei futás100 3 melluszás500 1 magasugrás 1

(1) Adjuk meg azt a versenyzőt, aki a legtöbb sportágban indult. Mi a neve, és hány sportágban indult?

(2) Van-e olyan iskola, melynek minden versenyzője legalább három sportágban indult? Ha van, adjuk meg az iskola azonosítóját is!

2 Specifikáció és algoritmus

2.1 Első feladatrész

Specifikáció:

$$A = (x: \text{infile}(\text{Eredmény}), \text{db}: \mathbb{N}, \text{név}: \mathbb{S}^*)$$

$$\text{Eredmény} = \text{rec}(\text{név}: \mathbb{S}^*, \text{id}: \mathbb{S}, \text{helyezések}: (\mathbb{S} \times \mathbb{N})^*)$$

$$\text{Versenyző} = \text{rec}(\text{név}: \mathbb{S}^*, \text{iskola}: \mathbb{S}, \text{db}: \mathbb{N})$$

Állapottér transzformáció:

$$A = (t: \text{enor}(\text{Versenyző}), \text{db}: \mathbb{N}, \text{név}: \mathbb{S}^*)$$

$$Ef = (t = t' \wedge |t| > 0)$$

$$Uf = (\text{db}, \text{elem} = \text{MAX}_{e \in t'} e.\text{db} \wedge \text{név} = \text{elem.név})$$

Visszavezetés: (maximum kiválasztás felsorolón)

$$E \rightarrow \text{Versenyző}$$

$$\text{max} \rightarrow \text{db}$$

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

$$(H, <) \rightarrow (\mathbb{N}, <)$$

Algoritmus:

```
t.first()
db := t.current().db
elem := t.current()
t.next()
while ¬t.end() do
  if db < t.current().db then
    db := t.current().db
    elem := t.current()
  end
  t.next()
end
név := elem.név
```

Table 1: (Felsoroló)

enor(Versenyző)	first()	next()	current()	end()
st: Status x: infile(Eredmény) e: Eredmény akt: Versenyző vég: \mathbb{L}	st,e,x: read next()	(...)	akt	vég

További 2 művelettel is rendelkezik a típus: *setEmptyRes()*, *setEmptyComp()*.
Ezek üresre állítják az *e* és az *akt* adattagot.

next():

$A = (\text{st: Status, x: infile(Eredmény), e: Eredmény, akt: Versenyző, vég: } \mathbb{L})$

$$Ef = (\text{st} = \text{st}' \wedge \text{x} = \text{x}' \wedge \text{e} = \text{e}' \wedge x_{id} \uparrow)$$

$$Uf = (\text{vég} = (\text{st} = \text{abnorm}) \wedge \neg \text{vég} \Rightarrow (\text{akt.név} = \text{e.név} \wedge \text{akt.db} = \sum_{i \in [1..|\text{e.helyezések}|]} 1)$$

Visszavezetés: (összegzés intervallumon)

$$\begin{aligned}
m &\rightarrow 1 \\
n &\rightarrow |\text{e.helyezések}| \\
s &\rightarrow \text{akt.db} \\
f(e) &\rightarrow 1 \\
(\mathbb{H}, +, 0) &\rightarrow (\mathbb{N}, +, 0)
\end{aligned}$$

Algoritmus:

```

setEmptyComp()
vég := st = abnorm
if  $\neg$  vég then
  akt.név := e.név
  akt.db := 0
  for  $i = 1..|\text{e.helyezések}|$  do
    | akt.db := akt.db + 1
  end
setEmptyRes()
st,e,x: read
end

```

2.2 Második feladatrész

Specifikáció:

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

$$\text{Eredmény} = \text{rec}(\text{név} : \mathbb{S}^*, id : \mathbb{S}, \text{helyezések} : (\mathbb{S} \times \mathbb{N})^*)$$

Állapottér transzformáció:

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

$$\text{Iskola} = \text{rec}(\text{id} : \mathbb{S}, \text{all} : \mathbb{L})$$

$$Ef = (t = t' \wedge t_{id} \uparrow)$$

$$Uf = (l, elem = \forall \mathbf{SEARCH}_{e \in v'}(all) \wedge l \Rightarrow id = elem.id)$$

Visszavezetés: (optimista lineáris keresés felsorolókra)

$$E \rightarrow Iskola$$

$$felt(e) \rightarrow all$$

Algoritmus:

```

l := false
t.first()
while  $\neg l \wedge \neg t.end()$  do
| elem := t.current()
| l := elem.all
| t.next()
| end
| if l then
| id := elem.id
| end

```

Table 2: (Felsoroló)

enor(Iskola)	first()	next()	current()	end()
st: Status x: enor(Versenyző) e: Versenyző akt: Iskola vég: \mathbb{L}	x.first() next()	(...)	akt	vég

next():

$A = (\text{st: Status, x: enor(Iskola), e: Versenyző, akt: Iskola, vég: } \mathbb{L})$

Versenyző = rec(név: \mathbb{S}^* , iskola: \mathbb{S} , db: \mathbb{N})

$Ef = (\text{st} = \text{st}' \wedge \text{x} = \text{x}' \wedge \text{e} = \text{e}' \wedge x_{id} \uparrow)$

$Uf = (\text{vég} = \text{x.vég}$

$\wedge \neg \text{vég} \Rightarrow (\text{akt.id} = \text{e.akt.iskola} \wedge \text{akt.all} = \text{igaz} \wedge \sum_{ee \in (x'.current(), x')} \bigwedge ee.iskola = \text{akt.id})$

Visszavezetés: (összegzés felsorolókra)

$t:\text{enor}(e) \rightarrow x:\text{enor}(\text{Versenyző})$

$s \rightarrow all$

$e \rightarrow ee$

$t' \rightarrow (x'.current(), x')$

$f(e) \rightarrow x.akt.iskola = \text{akt.id}$

$(H, +, 0) \rightarrow (\mathbb{L}, \wedge, true)$

Algoritmus:

vég := x.vég

if $\neg \text{vég}$ **then**

akt.id := x.current().iskola

akt.all := true

while $\neg x.end() \wedge x.current().iskola = \text{akt.id}$ **do**

 akt.all := akt.all \wedge (x.current().db ≥ 3)

 x.next()

end

end

3 Tesztelés

A felhasznált tételek: maximum kiválasztás, optimista lineáris keresés, összegzés.

3.1 Első feladat tesztesetei

- Nem létező fájl.
- Üres fájl.
- Egyetlen versenyző.
- Több versenyző.
- Első versenyző indult a legtöbb sportágban.
- Utolsó versenyző indult a legtöbb sportágban.
- Egy versenyző, 0 indulással.
- Egy versenyző, 1 indulással.
- Egy versenyző, 3 indulással.

3.2 Második feladat tesztesetei

- Nem létező fájl.
- Üres fájl.
- Egyetlen iskola.
- Több iskola.
- Első iskola minden versenyzője legalább 3 sportágban indult.
- Utolsó iskola minden versenyzője legalább 3 sportágban indult.
- Egy iskola, 1 versenyző, 0 indulással.
- Egy iskola, 1 versenyző, 1 indulással.
- Egy iskola, 1 versenyző, 2 indulással.
- Egy iskola, 1 versenyző, 3 indulással.