

Feladat

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 (szóközők nélküli sztring), majd halfajta-méret párok (szóközők nélküli sztring és természetes szám párok) formájában az adott versenyen elért fogásokat. Az adatok szóközőkkel vagy tabulátorjelekkel vannak egy soron belül elválasztva. A szöveges állomány sorait versenyek szerint rendezték. Feltehetjük, hogy a szöveges állomány helyesen van kitöltve.

Példa az állomány egy sorára: JANIBÁ Kiliti0512 Ponty 45 Ponty 53 Harcsa 96

(1) Adjuk meg ki és melyik versenyen fogta a legnagyobb harcsát! (Lehet, hogy senki sem fogott harcsát.)

(2) Soroljuk fel azokat a versenyeket, ahol harcsát is fogtak!

(1) Részfeladat megoldása

Főprogram terve:

Leegyszerűsítve a feladatot, elsősorban szeretnék egy olyan felsorolót létrehozni, ahonnan egy feltételes maximum kereséses függvénnyel kiválaszthatom a legnagyobb halat fogó horgászt. Ehhez szükségem van a horgász azonosítóra, a verseny azonosítójára, hogy fogott-e harcsát, és hogy mekkora volt a harcsa.

Specifikáció:

$$A = (t: \text{Angler}(\text{angler}), l: \mathbb{L}, \text{winner: angler})^1$$

$$\text{angler} = \text{rec}(\text{anglerId}: \mathbb{S}, \text{contestId}: \mathbb{S}, \text{isCatFish}: \mathbb{L}, \text{isNotCatFish}: \mathbb{L}, \text{sizeOfFish}: \mathbb{N})$$

$$Ef = (t = t_0)$$

$$Uf = (\text{angler}) = \mathbf{MAX}_{e \in_t, e.isCatFish} e.large \wedge$$

Feltételes maximum keresés:

$$t.enor(E) \sim t: \text{Angler}(\text{angler})$$

$$f(e) \sim e.large$$

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

¹ A kényelmesség kedvéért, kimenetre angler típusú adatot választottam. Tulajdonképpen az isNotCatFish adatra itt nem lesz szükségünk, ez a második feladathoz fog kelleni, amikor is azt kell megállapítani, hogy egy versenyen harcsát is fogtak, meg nem harcsa fajtájú halat.

winner.isCatFish=false;		
¬t.end()		
¬t.current().isCatFish	t.current().isCatFish∧ winner.isCatFish	t.current().isCatFish∧ ¬winner.isCatFish
—	winner.sizeOfFish < t.current().sizeOfFish	winner.isCatFish =true, winner.anglerId=
	winner.anglerId= t.current().anglerId, winner.contestId= t.current().contestId, winner.sizeOfFish=t.c urrent().sizeOfFish	— t.current().anglerId, winner.contestId= t.current().contestId, winner.sizeOfFish=t.current().si zeOfFish
t.next()		

Horgász felsorolója:

inputAngler= rec(anglerId:\$, contestId: \$, fish: : \$,large: N)

t:enor(angler) angler = rec(anglerId:\$, contestId: \$, isCatFish: \mathbb{L} ,sizeOfFish: N)

angler*	first()	next()	current() : angler	end() : \mathbb{L}
_x:infile(inputAngler) _dx : inputAngler _sx : Status _curr : angler _end : \mathbb{L}	next()	lásd külön	return _curr	return _end

Az enor(angler) first() és next() ugyanazok, és az alábbi feladatot kell megoldaniuk: olvassuk be a szöveges állomány (az f szekvenciális inputfájl) soron következő sorát. Ha ilyen nincs, akkor az _end változó értéke legyen igaz. Ha van, akkor ezt a sort sztringek és szám sorozatának tekintve kiszedhetjük belőle a horgász és a verseny azonosítóját, majd az ezeket követő részben megadhatjuk a legnagyobb méretű harcsát, amennyiben fogtak ilyen halat. Fontos, hogy itt több fogás is lehet egymás után, ezért erre különösen kell ügyelni, ezért nincs külön read(), mindent a next() függvény intéz a beolvasás tekintetében. Ezért a kapás adatait össze kell tömöríteni, méghozzá a legnagyobb kifogott harcsa értéke kerül oda, amennyiben van ilyen.

next() művelet

$A = (x:\text{infile}(\text{inputAngler}), dx:\text{inputAngler}, sx:\text{Status}, _curr:\text{angler}, _end:\mathbb{L}, in:\mathbb{S})$
 $Ef = (x = x' \wedge_{(\text{contestId})} \wedge dx = dx' \wedge sx = sx')$
 $Uf = ((_dx', (_sx', _dx', _x')) = (in = inline \wedge SELECT_{dx \in (_dx', _x')} (\neg x.\text{fail}() \wedge \neg line = ""))$
 $\wedge (_end = (_sx' = abnorm) = _x.\text{fail}() \vee line = ""))$
 $\wedge (\neg _end \rightarrow (_curr = \mathbf{MAX}_{_dx \in (_dx', _x')}^{in \gg _dx.\text{fish} \gg _dx.\text{sizeOfFish}^2} _dx.\text{sizeOfFish}) \wedge$
 $(_curr.\text{isCatFish} \rightarrow _dx.\text{fish} = "Harcsa")) \wedge (Ha \text{ létezik}$
 $_dx.\text{fish} \neq "Harcsa": _curr.\text{isNotCatFish} = true))^3$

Feltételes maximum keresés:

$t.enor(E) \sim x:\text{infile}(\text{inputAngler}) \text{ next}() \text{ felt: } in \gg _dx.\text{fish} \gg _dx.\text{sizeOfFish}$

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

$\text{felt}(e) \sim sx = abnorm \vee e.\text{isCatFish} \rightarrow _dx.\text{fish} = "Harcsa" \text{ }^4$

$sx = norm, in(inline)$				
$_curr.\text{isCatFish} = false, _curr.\text{isNotCatFish} = false$				
$\neg x.\text{fail}() \wedge \neg line = ""$				
$in \gg _dx.\text{fish}() \gg _dx.\text{sizeOfFish}$				$_sx = abnorm$
$\neg _dx.\text{fish} = "Harcsa"$	$_dx.\text{fish} = "Harcsa" \wedge _curr.\text{isCatFish}$	$_dx.\text{fish} = "Harcsa" \wedge \neg _curr.\text{isCatFish}$		
$_curr.\text{isNotCatFish} = true$	$_curr.\text{sizeOfFish} > _dx.\text{large}$			
	$_curr.\text{anglerId} = _dx.\text{anglerId}, _curr.\text{contestId} = _dx.\text{contestId}, _curr.\text{sizeOfFish} = _dx.\text{sizeOfFish}$	$_curr.\text{isCatFish} = true, _curr.\text{anglerId} = _dx.\text{anglerId}, _curr.\text{contestId} = _dx.\text{contestId}, _curr.\text{sizeOfFish} = _dx.\text{sizeOfFish}$		

² Tehát addig megy a ciklus, amíg van olyan elem a sorban, amit be kell olvasni. In a beolvasott sort reprezentáló istreamstring típusú adat, amelyből a >> jellel tároljuk el egy másik változóba a legközelebbi szóközig ott jelenlévő karaktereket.

³ Bár az isNotCatFish változó jelenleg nem kell, viszont a következő feladathoz jó szolgálatot tesz, és megspórol nekünk plusz beolvasást.

⁴ Ez azt is jelenti, hogy minden egyes meghíváskor csak egy sort olvasunk be, tehát a anglerID és a contentId ugyanazok egy next() futásán belül, a külső függvénnyel gondoskodunk arról, hogy a next() többszöri hívására végigmenjen a fájl.

$_end=(_sx==abnorm)^5$

(2) Részfeladat megoldása

Leegyszerűsítve a feladatot, itt egy olyan felsorolóra lesz szükségem, ahol meg tudom állapítani egy versenyről, hogy fogtak-e azon fogtak-e harcsát és nem harcsa fajtájú halat. Tehát szükségem van verseny azonosítókra és mindegyik versenyhez 2 boolean változóra.

Főprogram terve:

Specifikáció

$A = (t: \text{Contest}(\text{contest}), \text{contestId}: \mathbb{S})$

$Ef = (t = t_0)$

$Uf = (\text{contestId} = \bigoplus_{e \in t_0} \langle e \rangle)$

$\text{isCatfish} = \text{true} \wedge \text{isNotCatfish} = \text{true}$

Összegzés

$f(e) \sim \langle e \rangle$ ha $\text{isCatfish} = \text{true} \wedge \text{isNotCatfish} = \text{true}$

$s \sim \text{contestId}$

$H, +, 0 \sim \mathbb{S}, \bigoplus, \langle \rangle$

Verseny felsorolója:

Felsoroló:

$y: \text{enor}(\text{contest}) \quad \text{contest} = \text{rec}(\text{contestId}: \mathbb{S}, \text{isCatfish}: \mathbb{L}, \text{isNotCatfish}: \mathbb{L})$

contest*	first()	next()	current() : contest	end() : \mathbb{L}
$_curr : \text{contest}$ $_end : \mathbb{L}$ Angler $_tt$	$_tt.\text{first}()$ $\text{next}()$	lásd külön	return $_curr$	return $_end$

Mivel az előző feladatban létrehozott Angler felsoroló teljesen megfelelő adatokat olvas be, ezért nincs szükség itt további read beolvasásra, csak a megfelelő adatokat kell a már említett felsorólótól átvenni. A first() felhasználja az Angler osztály first() metódusát, és utána saját next() metódusát futtatja. Erre azért van szükség, hogy a kezdő értékek be legyenek olvasva a next számára. Éppen ezért a next egy first nélküli metódus. Feladata, hogy a már elállított értékekből, amely megmondja, hogy egy horgász, egy versenyen fogott-e harcsát, illetve nem harcsa fajtájú halat. Ezeket az információkat mind eltárolja, hogy az összefűzéskor a megfelelő értékek kerüljenek a sztringbe.

⁵ az $_sx$ változó különösebb hozzáadott értéket nem ad a függvényhez. Abban a sorban az $_end$ változó akár false értéket is kaphatna.

next() művelet

$A = (_tt: \text{Angler}(\text{angler}), _curr: \text{contest})$

$Ef = (_tt := _tt_0(\text{contestId}))$

$Uf = (_end = _tt.\text{end}() \wedge (\neg _tt.\text{end}() \rightarrow _curr.\text{contestId} = _tt.\text{contestId}$

$_tt_0.\text{contestId} = _curr.\text{contestId}$

$\wedge (_curr.\text{isCatfish} = \bigvee_{e \in (\text{tt}_0) _tt} _tt.\text{isCatFish})$

$_tt_0.\text{contestId} = _curr.\text{contestId}$

$\wedge (_curr.\text{isNotCatfish}) = \bigvee_{e \in (\text{tt}_0) _tt} _tt.\text{isNotCatFish})$

Két összegzés (vagyolás és vagyolás)

közös felsorolóval

$t: \text{enor}(E) \sim _tt: \text{Angler}(\text{angler}) \text{ first}() \text{ nélkül,}$

$\text{felt: } _dx.\text{contestId} = _curr.\text{contestId}$

$f(e) \sim _ _tt.\text{isCatFish} _tt.\text{isNotCatFish}$

$s \sim _curr.\text{isCatfish}, _curr.\text{isNotCatFish}$

$H, +, 0 \sim (\mathbb{L}, \mathbb{L}), (\vee, \vee), (\text{hamis}, \text{hamis})$

$_end := _tt.\text{end}(), _curr.\text{contestId} = _dx.\text{contestId}$
$\neg _tt.\text{end}() \wedge _curr.\text{contestId} = _tt.\text{contestId}$
$_curr.\text{isCatfish} = _curr.\text{isCatfish} \vee _tt.\text{isCatFish}$ $_curr.\text{isNotCatfish} := _curr.\text{isNotCatfish} \vee _tt.\text{isNotCatFish}$
$_tt.\text{next}()$

1. Beadandó/3. feladat

Teszttervezés

A megoldásban 3 programozási tételt alkalmaztunk: kiválasztás, feltételes maximum keresés, összegzés

Első feladat kiválasztás és feltételes belső maximum keresés tesztesetei:

- intervallum hossza szerint :
 1. Üres állomány.
 2. Egyetlen horgász.
 3. Több horgász.
- intervallum eleje és vége szerint:
 4. Első horgász fogta a legnagyobb harcsát.
 5. Az utolsó horgász fogta a legnagyobb harcsát.
- tétel szerint:
 1. Van megfelelő fogás és horgász.
 2. Nincs megfelelő fogás és horgász.
 3. Több megfelelő fogás is van.
 1. Olyan sor, amely nem tartalmaz fogásokat.
 2. Olyan sor, amely egy fogást tartalmaz.
 3. Olyan sor, amely több fogást is tartalmaz.

Első feladat kiválasztás és feltételes külső maximum keresés tesztesetei:

- intervallum eleje és vége szerint:
 4. Első horgász fogta a legnagyobb harcsát.
 5. Az utolsó horgász fogta a legnagyobb harcsát.
- tétel szerint:
 1. Több megfelelő horgász is van.

Első feladat kiválasztás és feltételes belső összegzés tesztesetei:

- intervallum hossza szerint :
 1. Üres állomány.
 2. Egyetlen horgász.
 3. Több horgász.

1. Beadandó/3. feladat

– intervallum eleje és vége szerint:

4. Első versenynél fogtak harcsát.
5. Az utolsó versenynél fogtak harcsát.

– tétel szerint:

1. Van megfelelő verseny.
 2. Nincs megfelelő verseny.
 3. Több megfelelő verseny is van.
1. Olyan sor, amely nem tartalmaz fogásokat.
 2. Olyan sor, amely egy fogást tartalmaz.
 3. Olyan sor, amely több fogást is tartalmaz.