**Tóth Botond**  **2.beadandó/22.feladat** 2021.04.09.

MQH41V

[mqh41v@inf.elte.hu](mailto:mqh41v@inf.elte.hu) vagy tothbotond00@gmail.com

10. csoport

**Feladat**

Egy általános iskola alsó tagozatán papírgyűjtő versenyt rendeztek. A verseny 2018. szeptember 1-től, december 31-ig tartott. Feljegyezték a verseny adatait, és egy szöveges állományban tárolták el. A fájl egy sorának felépítése: elsőként a tanuló neve szerepel (két vagy több szóközök nélküli sztring), majd az osztálynak az azonosítója (1-4 számjeggyel kezdődő, szóközt nem tartalmazó sztring, például 1a, 2b, 4c), majd a papírgyűjtés adatai: dátum-súly (a dátum EEEE/HH/NN alakú sztring, a súly egy pozitív valós szám: a gyűjtött papír súlya kilogrammban megadva) formájában. A sor dátum szerint rendezett. Az adatok szóközökkel vagy tabulátorjelekkel vannak egy soron belül elválasztva. A szöveges állomány sorait osztály-azonosító szerint rendezték. Feltehetjük, hogy a szöveges állomány helyesen van kitöltve. Példa az állomány egy sorára:

Nagyon Szorgalmas Eszter 4c 2018/09/10 4.5 2018/09/22 3.5 2018/11/05 1.2

(1) Melyik tanuló gyűjtötte a legtöbb papírt? Adjuk meg a nevét, az osztályát, és hogy mennyit gyűjtött!

(2) Van-e olyan osztály, hogy mindegyik tanulója (aki a versenyben részt vett) legalább 10 kg papírt gyűjtött a verseny során? Ha igen, adjuk meg az osztályt is!

**(1)Részfeladat megoldása:**

**Főprogram terve:**

A = (f:infile(Sor), max:R(valós),elem:Megoldas)

Sor = rec(nev:String, osztaly:String, adatok : Adat\*)

Adat = rec(datum:String, mennyiseg:R(valós))

Papir = rec(nev: String, osztaly: String, kilo: R)

Új Állapottér:

A = (t.enor(Papir), max:R(valós), elem: Papir)

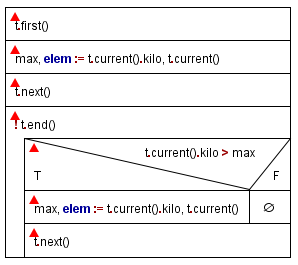
Ef = (t = t’ ∧|t| > 0)

Uf = (max, elem = MAX e∈𝑡′ e.kilo )

Visszavezetés: maximum keresés, egyedi felsorolón

E ~ Papir

H ~ R

 f(e) ~ e.kilo

**Papir egyedi felsoroló:**

|  |  |
| --- | --- |
| enor (Papir) | first(),next(),current(),end() |
| f:infile(Sor)  cur: Papír  end: L | first() ~ next()  next() ~ lentebb…  current() ~ **return** cur  end() ~ **return** end |

Státusz = {norm, abnorm}

Az enor(Papir) egyedi felsorolónak az alábbi feladatot kell megoldania: a txt fáljunknak (szekvenciális inputfálj) soron következő sorát kell beolvasnia. Ha nincs ilyen az end változó értékét igazra kell állítanunk. Különben el kell tárolunk a cur változóba a diáknak a nevét, osztályát, majd a következő adatokban összegeznünk kell a diák által gyűjtött hulladék mennyiségét és végül azt is a cur változóba kell tenni. Mivel soronként kell haladnunk ezért a first() megegyezik a next()-el.

Anext = (f:infile(Sor), cur:Papir, end:L)

Efnext = (f = f’)

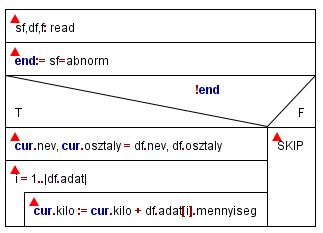
Ufnext = (sf,df,f = read(f’) ∧ end = (sf = abnorm) ∧ ¬ end -> cur.nev = df.nev ∧ cur.osztaly = df.osztaly ∧ cur.kilo = ∑ 𝑖∈[1..|𝑑𝑓.adat|] df.adat[i].mennyiseg)

Viszavezetés: összegzés, intervallum felsorolón

t:[m..n] ~ i:[1..|df.adat| ]

v[i] ~ df.adat[i].mennyiseg

s ~ cur.kilo



**(2)Részfeladat megoldása**

**Főprogram terve**

A = (f:infile(Sor), l: L, osztaly : String)

Sor = rec(nev:String, osztaly:String, adatok : Adat\*)

Adat = rec(datum:String, mennyiseg:R(valós))

Új állapottér:

A = (t:enor(Osztaly), l:L , osztaly: String)

Osztaly = rec (osztaly:String, szorgos: L)

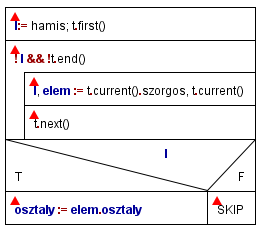
Ef = (t = t’)

Uf = (l, elem = SEARCH e∈𝑡 ′(e.szorgos) ∧ l-> osztaly = elem.osztaly)

Visszavezetés: lineáris keresés, egyedi felsorolón

E ~ Osztaly

felt(e) ~ e.szorgos



**Osztaly egyedi felsorló**

|  |  |
| --- | --- |
| enor(Osztaly) | first(), next(), current(), next() |
| tt:enor(Papir)  cur: Osztaly  end: L | first() ~ tt.first()  next() ~ lentebb…  current() ~ **return** cur  end() ~ **return** end |

Státusz = {norm, abnorm}

Osztaly = rec (osztaly:String, szorgos: L)

Ebben a feladatban fel tudjuk használni az 1. feladatban létrehozott felsorolót. Abban sikeresen meg tudtuk számolni, hogy egy diák mennyi gyűjtött, így egy optimista eldöntéssel osztályonként is meg tudjuk nézni, hogy minden diák 10 kg-nál többet gyűjtött-e. A next() művelet csak addig fog tartani míg ugyanannak aza osztálynak a diákjait vizsgáljuk, ha netalántán másik osztályt nézünk már akkor a next() leáll. Fontos különbség itt, hogyha találtunk már olyan adatot ami miatt nem lesz igaz az eldöntés, folytatnunk kell az algoritmust, hiszen nem tudjuk pontosan melyik sortól kezdődik a következő osztály (így inkább összegzésre tudjuk visszavezetni).

Anext = (tt:enor(Papir), cur: Osztaly, end:L )

Efnext= (tt = tt’)

Ufnext= (end =tt’.end() ∧ ¬end -> (cur.osztaly=tt’.current().osztaly ∧ cur.szorgos = igaz ∧ (cur.szorgos,tt) = ) )

Viszavezetés: összegzés (optimista eldöntés), intervallum felsorolón

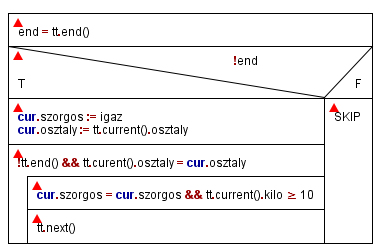
t:enor(E) ~ tt:enor(Papir)

amíg e.osztaly = cur.osztaly (first nélkül)

eϵt’ ~ e ϵ (tt’.current(),tt’)

s ~ cur.szorgos

(H,+,0) ~ (L, ∧, igaz)



**Tesztelési Terv**

A feladat megoldásában maximum keresést, összegzést, lineáris keresést és összegzésre visszavezetett optimista eldöntést használtunk.

1. Első feladat maximum keresésének tesztesetei:
   1. Intervallum hossza szerint
      1. Üres állomány
      2. Egy elemet tartalmazó állomány
      3. Több elemet tartalmazó állomány
   2. Maximum helye szerint
      1. Első helyen van a maximum
      2. Végén van a maximum
      3. Random helyen van a maximum
   3. Maximum tétel alapján
      1. Csak egy maximum van
      2. Több maximum is van (hamarabb beolvasott elem lesz a maximum)
2. Egy diák a beadott papírjainak tömegének összegének (összegzés) tesztesetei:
   1. Intervallum hossza szerint és a tétel szerint
      1. Sor amiben nincs számolandó adat
      2. Sor amiben egy számolandó adat
      3. Sor amiben sok összegzendő adat van
3. Második feladat lineáris keresésének tesztesetei:
   1. Intervallum hossza szerint
      1. Üres állomány
      2. Egyetlen osztály
      3. Több osztály is
   2. Keresett osztály helye szerint
      1. Első osztály szorgos
      2. Csak az utolsó osztály szorgos
   3. Keresés tétel szerint
      1. Van szorgos osztály
      2. Nincs szorgos osztály
      3. Több szorgos osztály is van
4. Egy osztály szorgosságának ledöntése (optimista eldöntés) tesztesetei:
   1. Intervallum hossza szerint
      1. Egy ember jár az osztályba
      2. Több ember jár az osztályba
   2. Nem megfelelő elem helye szerint
      1. Első ember nem gyűjtött 10 kilót
      2. Utolsó ember nem gyűjtött 10 kilót
      3. Random ember nem gyűjtött 10 kilót
   3. Eldöntés tétel szerint
      1. Egyik gyerek sem gyűjtött 10 kilót
      2. Egy gyerek nem gyűjtött 10 kilót
      3. Minden gyerek 10 kilót gyűjtött