

- I. Tudjuk, hogy van olyan $f(i)$ és $g(j)$ érték, hogy $f(i)+g(j)=e$, ahol $f,g : \mathbb{N} \rightarrow \mathbb{R}$ és $e \in \mathbb{R}$. Adjunk meg ilyen i -t és j -t!

Gondolatban rendezzük el az $f(i)+g(j)$ értékeket egy végtelen kiterjedésű 0-tól kezdődő indexelésű mátrixba úgy, hogy a mátrix i -dik sorának j -edik eleme az $f(i)+g(j)$ érték legyen. A feladat megoldásához ezen mátrix elemeit kell bejárni, de úgy, hogy véges lépésen belül bármelyik eleméhez el lehessen jutni. Erre a klasszikus (sorfolytonos) bejárás nem alkalmas, mert az soha nem fog a mátrix első (0.) során kívül más elemhez eljutni, lévén a sor végtelen hosszú. Definíáljunk egy olyan egyedi felsorolást erre a mátrixra, amely a mellékátlók mentén járja be az elemeket. Egy t felsorolótól tehát index párokat várunk $((i,j) \in t)$, hogy ezek segítségével kiszámolhassuk az $f(i)+g(j)$ értéket.

A feladat specifikációja és absztrakt programja egyedi felsorolóval:

$A = (t: \text{enor}(\mathbb{N} \times \mathbb{N}), e: \mathbb{R}, i: \mathbb{N}, j: \mathbb{N})$

$Ef = (e = e')$

$Uf = (e = e' \wedge (i,j) = \text{SELECT}_{(i,j) \in t} (f(i)+g(j)=e))$

| | |
|--|----------|
| t.First() | |
| $f(t.Current().i) + g(t.Current().j) \neq e$ | |
| | t.Next() |

A felsoroló megvalósításához azt kell látni, hogy

- a reprezentálásához szükség lesz az i, j indexekre,
- a felsorolást a 0,0 indexekkel kell kezdeni (First()),
- az indexek felsorolásával a mellékátlók mentén haladjunk tovább (Next()),
- az aktuális indexpárt adja vissza a Current(),
- End() művelet most nem kell, a terminálást a kiválasztás tétele garantálja.

enor($\mathbb{N} \times \mathbb{N}$)

| $(\mathbb{N} \times \mathbb{N})^*$ | First() | Next() | Current() |
|------------------------------------|----------------|---|-----------|
| $i, j: \mathbb{N}$ | $i, j := 0, 0$ | ha $i > 0$ akkor $i, j := i-1, j+1$ ha $i = 0$ akkor $i := j+1 ; j := 0$ | i, j |

Összeolvasva

| | |
|---|------------|
| $i, j := 0, 0$ | |
| $f(i) + g(j) \neq e$ | |
| <div style="border: 1px solid black; padding: 5px; text-align: center;"> $i > 0$ </div> | |
| $i, j := i-1, j+1$ | $i := j+1$ |
| | $j := 0$ |

- II. Ismerjük a Föld felszín tengerszint feletti magasságait egy adott szakaszon, amelyeket egy szekvenciális inputfájlban tároltunk el. Hány horpadás van a megadott felszínen?

$A = (f : \text{infile}(\mathbb{R}), \text{db} : \mathbb{N})$

$Ef = (f = f')$

A szekvenciális fájl klasszikus felsorolása egyszerre csak egy elemét mutatja a fájlnek, nekünk azonban egyszerre három egymást követő elemre kell hivatkoznunk. Olyan felsorolás kell, amely ezt biztosítja.

enor $(\mathbb{R} \times \mathbb{R} \times \mathbb{R})$

| $(\mathbb{R} \times \mathbb{R} \times \mathbb{R})^*$ | First() | Next() | End() | Current() |
|--|---|---|-------------|---------------|
| elő, akt, köv : \mathbb{R} f: infile(\mathbb{R}) st : Status | st, elő, f : read st, akt, f : read st, köv, f : read | elő := akt akt := köv st, köv, f : read | st = abnorm | elő, akt, köv |

A feladat specifikációja és absztrakt megoldása az egyedi felsorolóval:

$A : (t : \text{enor}(\mathbb{R} \times \mathbb{R} \times \mathbb{R}), \text{db} : \mathbb{N})$

$Ef : (t = t' \wedge e = e')$

$Uf : (\text{db} = \sum_{\substack{(\text{elő}, \text{akt}, \text{utó}) \in t' \\ \text{elő} > \text{akt} \wedge \text{akt} < \text{utó}}} 1)$

| | |
|---|---|
| db := 0 | |
| t.First() | |
| ¬t.End() | |
| <div style="border: 1px solid black; padding: 5px; text-align: center;"> t.Current().elő > t.Current().akt ∧ t.Current().akt < t.Current().köv </div> | |
| db := db+1 | — |
| t.Next() | |

Összeolvasva

| | |
|---|---|
| db := 0 | |
| st, elő, f : read; st, akt, f : read; st, köv, f : read | |
| sf = norm | |
| <div style="border: 1px solid black; padding: 5px; text-align: center;"> elő < akt ∧ akt > köv </div> | |
| db := db+1 | — |
| elő := akt; akt := köv; st, köv, f : read | |

- III. Adottak a szekvenciális inputfájlban a hallgatók évközben szerzett jegyei, a hallgatók neve szerint rendezett formában. Számoljuk ki az egyes hallgatók átlagait!

$A : (x : \text{infile}(\text{Hallg}), y : \text{Outfile}(\text{Hallg})) \quad \text{Hallg} = \text{rec}(\text{azon} : \text{String}, \text{jegy} : \mathbb{R})$
 $Ef : (x = x_0 \wedge x \nearrow_{\text{azon}})$

A feladat megoldásához nem az x szekvenciális inputfájl felsorolását kell közvetlenül felhasználni: olyan egyedi felsoroló kell, amely képes hallgatónként felsorolni az átlagokat, azaz lényegében előállítja a listázandó adatokat.

A feladat specifikációja és absztrakt megoldása tehát egyszerű: ez egy másolás.

$A = (t : \text{enor}(\text{Hallg}), y : \text{Outfile}(\text{Hallg}))$
 $Ef = (t = t_0)$
 $Uf = (y = t_0) = (y = \oplus_{h \in t_0} \langle h \rangle)$

| |
|--|
| $y := \langle \rangle$ $t.\text{First}()$ |
| $\neg t.\text{End}()$ |
| $y:\text{write}(t.\text{Current}())$ |
| $t.\text{Next}()$ |

enor(Hallgató)

| Hallgató* | First() | Next() | End() | Current() |
|--|--|---|---------------|-----------|
| $h : \text{Hallg}$ $\text{vége} : \mathbb{L}$ $x : \text{infile}(\text{Hallg})$ $dx : \text{Hallg}$ $sx : \text{Status}$ | „első hallgató” $sx, dx, x : \text{read}$ $\text{Next}()$ | „következő hallgató” lásd lenn | vége | h |

Next()

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

$Uf : (\text{vége} = (sx' = \text{abnorm}))$

$\wedge (sx' = \text{norm} \rightarrow h.\text{azon} = dx'.\text{azon})$

$\wedge (\text{össz}, db), (sx, dx, x) = \sum_{(sx, dx, x) \in (dx', x')}^{sx = \text{norm} \wedge dx.\text{azon} = h.\text{azon}} (dx.\text{jegy}, 1)$

$\wedge h.\text{jegy} = \text{össz}/db$

Megj.: Látjuk, hogy az egyedi felsoroló megvalósítása az x szekvenciális inputfájl szokásos felsorolására támaszkodik. Az x felsorolásának $\text{Next}()$ művelete az $sx, dx, x: \text{read}$, a $\text{Current}()$ művelete a dx -et adja vissza. Az összegzés esetében (itt számpárokat összegzünk, és a neutrális elem a $(0,0)$ számpár) azonban kicsit módosul a szekvenciális inputfájl standard felsorolása. Egyrészt a már korábban kiolvasott dx' elemet is fel kell dolgozni az x' elemeivel együtt, tehát itt a dx' , majd az x' elemeinek felsorolásáról van szó. Erre utal a $(sx, dx, x) \in (dx', x')$ jelölésben szereplő (dx', x') , amely mintha újra egybe forrasztaná egy pillanatra a korábban már kiolvasott dx' elemet az x' sorozattal (fájllal). Ennek az a következménye, hogy nem kell (nem szabad) előre olvasással kezdeni a (dx', x') felsorolását, hiszen a felsorolás első eleme, a dx' már a kezünkben van. A felsorolás $\text{First}()$ művelete tehát az üres utasítás lesz. Másrészt a rekordok olvasása nem a fájl végéig tart, hanem csak addig, amíg $sx = \text{norm} \wedge dx.\text{azon} = h.\text{azon}$. Ennek alapján tehát módosul a felsorolás $\text{End}()$ művelete.

t.Next()

| | |
|--------------------------------|---|
| vége := sx = abnorm | |
| ¬vége | |
| h.azon := dx.azon | — |
| össz, db := 0, 0 | |
| sx=norm ∧ dx.azon = h.azon | |
| össz, db := össz+dx.jegy, db+1 | |
| sx, dx, x : read | |
| h.jegy := össz/db | |