Programozás 2.beadandó

Hartyányi Kevin

2018.03.31

Contents

List of Tables

1 Feladat

Egy szöveges állományban a Formula 1 autóverseny ez évi bajnokságának eddigi eredményeit tároljuk csapatok, azon belül versenyzők szerint rendezett formában. (Egy csapat két versenyzőt futtat.) Az állomány minden sorában egy versenyzőnek egy versenyen elért helyezése található. Egy sor adatai: a versenyző neve (sztring), a csapatának neve (sztring), verseny helyszíne (sztring), helyezés (pozitív egész szám), szóközökkel és/vagy tabulátor jelekkel vannak elválasztva. Melyik versenyző vezeti jelenleg a bajnokságot?

Feltehetjük, hogy az állomány sorai helyesen vannak kitöltve. Egy versenyen az elért helyezésük alapján az alábbi pontszámot kapják a versenyzők: `www.formula1.com/inside_f1/rules_and_regulations/sporting_regulations/8681/`

Table 1: Pontszámok

Helyezés	**Pontszám**
1st:	25
2nd:	18
3rd:	15
4th:	12
5th:	10
6th:	8
7th:	6
8th:	4
9th:	2
10th :	1

2 Specifikáció

$A = (x : \mathit{infile}(R), n : \mathbb{C}h^*)$

$R = \mathit{rec}(\text{név: } \mathbb{C}h^*, \text{ csapat: } \mathbb{C}h^*, \text{ hely: } \mathbb{C}h^*, \text{ helyezés: } \mathit{int})$

$\mathit{Ef} = (x = x' \wedge |x| \geq 1 \wedge \forall i[1 \ldots x-1] : (x_i.\text{név} \leq x_{i+1}.\text{név} \wedge x_i.\text{név} = x_{i+1}.\text{név} \Rightarrow x_i.\text{csapat} \leq x_{i+1}.\text{csapat}))$

Azonban így csak nagyon bonyolultan tudnánk felírni az utófeltételt. Ezért a feladat szempontjából érdemesebb egy olyan felsoroló objektumra megfogalmazni, mely képes összegezni egy adott játékos által elért összpontot.

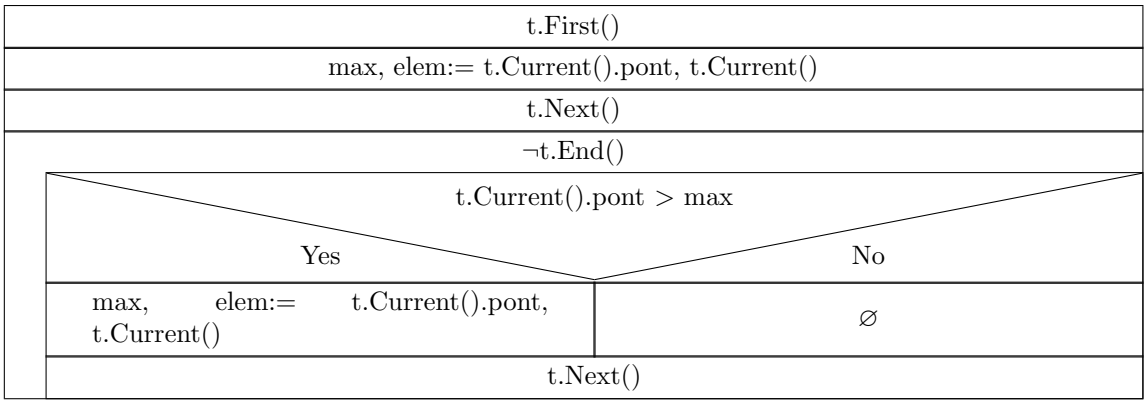
$$A = (t : \mathit{enor}(PAR), n : \mathbb{C}h^*)$$
$$\mathit{Ef} = (t = t' \wedge |t| > 0)$$
$$\mathit{Uf} = ((-, (\mathit{nev}, -)) = \underset{\mathbf{e} \in \mathbf{t'}}{\mathrm{MAX}}\, e.\mathit{pont})$$

2.1 Visszavezetés

Az utófeltételből látszik, hogy maximumkiválasztást kell használnunk.
PAR = (név:$\mathbb{C}h^*$, pont: int)

Table 2: Visszavezetés

Tétel	Feladat
E	PAR
$\beta : E \longrightarrow H$	$f : PAR \longrightarrow \mathbb{N}$ $f(e) = e.pont$
max	-
elem	(n, -)



2.2 Felsoroló

Table 3: Felsoroló

enor(PAR)	first(), next(), current(), end()
x: infile(String), dx: R, sx: Státusz	first(): sx, dx, x: read; next() next(): külön
akt: PAR	current(): akt
vége: $\mathbb{L}$	end(): vége

A next()-nek az alábbi műveletet kell végrehajtania:

Adott egy szöveges állomány, melynek minden sorában egy versenyző adatai szerepelnek, melyek közül már kiolvastuk az első sort (ez van a dx változóban, ha a kiolvasás nem volt sikeres, akkor sx = abnorm) és a következő sorokat kell feldolgozni. Ha sx = norm akkor elkezdi a feladatot végrehajtani a next, különben a vége változót igazra állítja.

Először az akt változó név részét beállítjuk a dx változó név részére és az akt összpont részét nullára állítjuk. Ezután a már korában látott helyezés-pontszám táblázatból megnézzük, hogy az adott versenyen mennyi pontot ért el és az eredményt hozzáadjuk az akt összpont részéhez, végül beolvassuk a következő sort és ha az nem a fájl vége és dx név része megegyezik az előbb beállított akt név részével, (tehát a sorban ugyan arról a versenyzőről van információ) akkor a elvégezzük az előbb említett folyamatot és ezt addig folytatjuk amíg a feltétel igaz.

2.3 Next() specifikációja

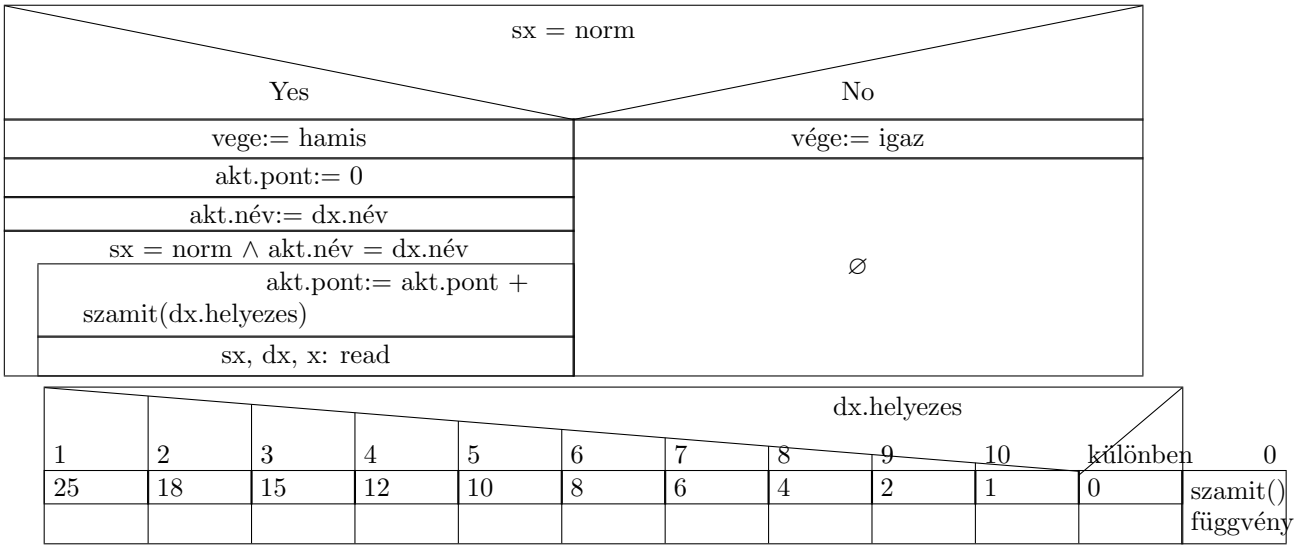
$$A^{next} = (x : infile(String), dx : R, sx : Status, vege : \mathbb{L}, akt : PAR)$$

$$Ef^{next} = (x = x' \wedge dx = dx' \wedge sx = sx' \wedge akt.pont = 0 \wedge akt.nev = dx.nev)$$

$$Uf^{next} = (vege = \neg(sx = norm) \wedge \neg(vege \longrightarrow \sum_{dx \in x}^{dx=norm \wedge akt.nev=dx.nev} akt.pont = akt.pont + szamit(dx.helyezes)))$$

$$szamit(x) = \begin{cases} 25 & \text{ha x = 1} \\ 18 & \text{ha x = 2} \\ 15 & \text{ha x = 3} \\ 12 & \text{ha x = 4} \\ 10 & \text{ha x = 5} \\ 8 & \text{ha x = 6} \\ 6 & \text{ha x = 7} \\ 4 & \text{ha x = 8} \\ 2 & \text{ha x = 9} \\ 1 & \text{ha x = 10} \\ 0 & \text{különben} \end{cases}$$

2.4 Next() struktogramja



dx.helyezes											
1	2	3	4	5	6	7	8	9	10	különben	0
25	18	15	12	10	8	6	4	2	1	0	szamit() függvény

3 Implementáció

3.1 Program váz

Table 4: Program váz

main.cpp	**list.h**	**listenor.h**	**listenor.cpp**
int main()	struct R	enum Status	void read()
	struct PAR	class Listenor	void next()
		void First()	Listenor()
		bool End()	
		PAR current()	

3.2 Felsoroló osztálya

```
class Listenor
{
        public:
                enum Status{abnorm, norm};
                enum Exceptions{FILEERROR};

                Listenor(std::string filename);
                void First() {read(); Next();};
                void Next();
                Scores Current() const {return akt;};
                bool End() const {return end;};


        private:
                bool end;
                void read();
                Scores akt;
                List dx;
                Status sx;
                std::ifstream x;
};
```

A szöveget a getline(x, n) utasítással olvassuk be soronként, melyet a read() műveletbe helyezünk el, itt n stringstream típusú. Az olvasás után beállítjuk az sx értékét, és ha az olvasás sikeres, akkor feltöltjük a dx változót az n változóból.

3.3 Tesztelési terv

A programban egy összegzés és egy maximum-kíválasztás tételt használtunk fel.

1. Összegzés alapján
   (a) itervallum hossza szerint:
      i. Üres fájl
      ii. Csak üres sorokat tartalmazó fájl
      iii. Egyetlen csapatott tartalmazó fájl
      iv. Sok csapatot tartalmazó fájl
   (b) intervallum eleje és vége szerint:
      i. Több csapatott tartalmazó fájl, ahol az első versenyzőnek 0 pontja van
      ii. Több csapatott tartalmazó fájl, ahol az utolsó versenyzőnek 0 pontja van
      iii. Több csapatott tartalmazó fájl, ahol az egyik csapatban az egyik versenyzőnek 0 pontja van, míg a másiknak van.
2. Maximum-kiválasztás alapján

(a) itervallum hossza szerint:
    i. Kevés versenyzőt tartalmazó állomány
    ii. Sok versenyzőt tartalmazó állomány
    iii. Minden versenyzőnek ugyanannyi pontja van
    iv. Minden versenyzőnek különböző pontja van
(b) intervallum eleje és vége szerint:
    i. Az első versenyzőnek van maximum pontja
    ii. Az utolsó versenyzőnek van maximum pontja
    iii. Több versenyzőnek van maximum pontja