

Programozás beadandó feladat:
2. beadandó / 7. feladat

Készítette: Podlovics Péter Dávid
Neptun-kód: D0W5T4
E-mail: peter.d.podlovics@gmail.com

Gyakorlatvezető neve: Veszprémi Anna/Hudoba Péter

2016. május 6.

Contents

Feladat:	3
Megoldási terv:	3
Specifikáció:	3
Visszavezetés:	3
Maximum kiválasztás:	3
Megszámlálás:	Error! Bookmark not defined.
Implementáció:	8
Az adattípusok megvalósítása:	Error! Bookmark not defined.
Bemenő adatok formája:	8
Program váz:	8
Tesztelési terv:	10
Fekete doboz tesztesetek:	10
Érvényes tesztesetek:	10
Érvénytelen tesztesetek:	Error! Bookmark not defined.

Feladat:

Egy szöveges állományban bekezdésekre tördelt szöveg található. Egy bekezdés egy vagy több nem üres sorból áll. A bekezdéseket üres sorok vagy az állomány eleje illetve vége határolja. Melyik a leggazdagabb bekezdés, azaz hányadik az a legalább három soros bekezdés, ahol a legnagyobb a szavak számának és a sorok számának hányadosa? (A szövegben egyik szó sincs több sorra törölve.) Tipp: Készítsen olyan felsorolót, amelyik a bekezdéseket, pontosabban azoknak számunkra fontos tulajdonságát sorolja fel!

Megoldási terv:

Specifikáció:

$$A = (t: \text{StreamEnor}(\text{Paragraph}), l \in L, \text{elem is Paragraph}, \text{max} \in \mathbb{R})$$
$$Ef = (t = t')$$

$$Uf = ((l, \text{max}, \text{elem}) = \underset{e \in t'}{\text{Max}} (f(e))$$
$$\text{hasThreeLines}(e)$$

$$\text{Paragraph} = \text{rec}(id \in \mathbb{N}, \text{numOfWords} \in \mathbb{N}, \text{numOfRows} \in \mathbb{N})$$

$$f: \text{Paragraph} \rightarrow \mathbb{R}$$
$$f = \frac{x.\text{numOfWords}}{x.\text{numOfRows}}$$

Visszavezetés:

Feltételes maximum keresés:

$$e \sim \text{StreamEnor}(\text{Paragraph})$$

$$\beta(x) \sim \text{hasThreeLines}(x)$$

$$< \sim f(p1) < f(p2)$$

A maximum keresést kiválasztást a `condMaxSearch` függvény fogja végrehajtani

condMaxSearch

I:=hamis, t.first()		
!t.end()		
hasThreeLines(t.current())		
True && I		True && !I
f(t.current()) > max		False
I	H	
elem := t.current() max := f(t.current())	Ø	I := True elem := t.current() max := f(t.current()) Ø
t.next()		

StreamEnor:

StreamEnor(E)	First()	Next()	l:= End()	e := Current()
f:infile(E) sf : Status df : E read	Sf,df,read	Sf,df,read	Sf = abnorm	e := df

A read függvény eltér a már megszokott soronkénti olvasástól. Ez a függvény egyszerre egy teljes Paragraph-ot beolvas az adott streamről. Ezt a következő képpen teszi:

Beolvassa az összes üres sort a stream elejéről, majd elkezdi olvasni az aktuális paragrafus sorait, és ezeket fogja feldolgozni. Mikoözben olvassa a sorokat, folyamatosan módosítja a jelenlegi Paragraph numOfWorks és numOfRows mezőit.

$$A^{read} = (h: infile(String), line \in String, p \in Paragraph, isEndOfFile \in L, st \in Status)$$

$$EF^{read} = (h = h^1 \wedge line = line^1 \wedge st = st^1 \wedge p = p^1)$$

$$UF^{read} = (st^2, line^2, h^2 = \text{Select } (st = abnorm \vee line =) \wedge line \in (sor^1, h^1))$$

$$isEndOfFile = (st^2 = abnorm) \wedge (\neg isEndOfFile \Rightarrow p.id = p.id + 1 \wedge$$

$$p.numOfRows, st, line, h = \sum_{\substack{line \neq "" \\ line \in (sor^2, h^2)}} 1 \wedge$$

$$p.numOfWorks, st, line, h = \sum_{\substack{line \neq "" \\ line \in (sor^2, h^2)}} words(line)))$$

$$words(line) = \sum_{word \in line} 1$$

StreamEnor:read()

line = "" && st = norm	
st,line,h:read	
isEndOfFile := st = abnorm	
I	H !isEndOfFile
<p>p.id := p.id + 1 p.numOfRows := 0 p.numOfWords := 0</p>	Ø
line.length() > 0	
<p>p.numOfRows := p.numOfRows + 1 actWords := words(line) p.numOfWords := p.numOfWords + actWords</p>	
st,line,h:read	

actWords := words(line)

actWords := 0

st,word,line:read

st = norm

actWords := **actWords** + 1

st,word,line:read

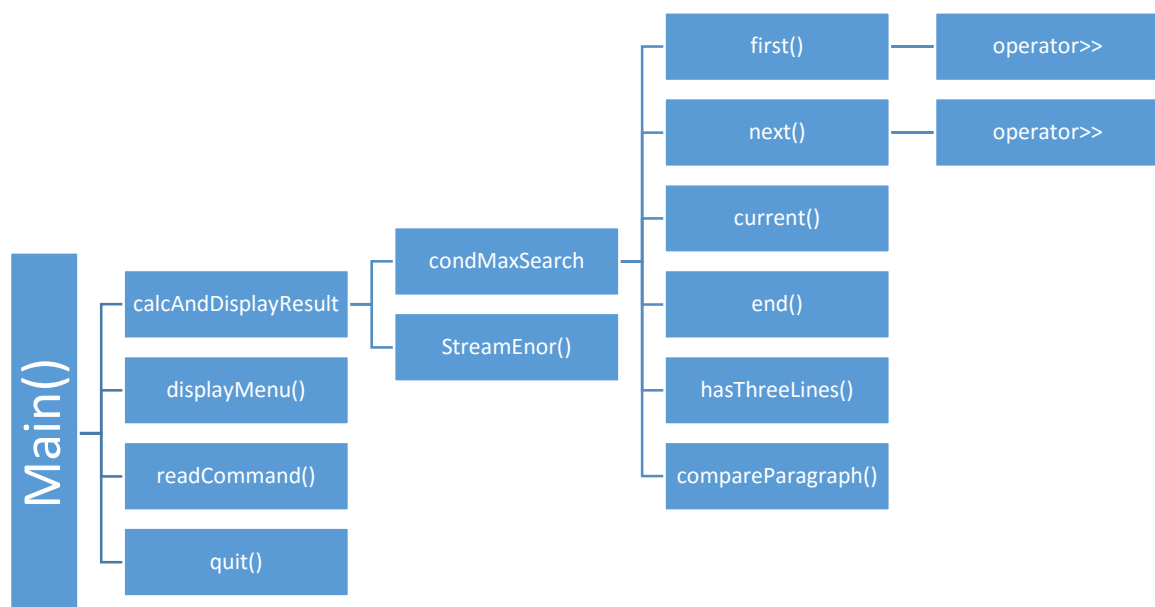
Implementáció:

Bemenő adatok formája:

A bemenő adatokat kétféle képpen adhatjuk meg: konzolról való beolvasás során, illetve szöveges állományból való beolvasással. A konzolról történő adatbevitel során a bemenet megadása után `end_of_file` jelet kell küldenünk a programnak (ez operációs rendszerenként eltérő lehet)

Program váz:

A függvények kapcsolódási rendszere:



A Paragraph-ok beolvasását egy overload-olt `operator>>()` végzi. A soronkénti olvasásért a `getline()` függvény felelős. Egy sort egy `stringstream`-ben tárolunk, ezt dolgozzuk fel.

A program állományai:

Main.cpp	Menu.cpp	Paragraphe.cpp	Enor.h	Enor.hpp	Menu.h	Paragraphe.h
Struct QueryResult, Bool hasThreeLines(), Bool compareParagraphs(), QueryResult condMaxSearch()	Int checkedReadLine(), Void displayMenu, Bool checkCommand, Int readCommand, Void quit()	Istream& operator >>, Ostream& operator <<	Enum Status, Struct StreamEn orExcept ion, Class Enor, Class StreamEn or	StreamEnor Exception(), Ostream& operator<< , StreamEnor ()	A menu.cpp függvényei	Class Paragraphe

Tesztelési terv:

Fekete doboz tesztesetek:

Érvényes tesztesetek:

1) A bekezdések szintjén (max. ker)

a) Intervallum hossza szerint:

i) Üres állomány:

Bemenet: in1.txt

Kimenet: NOT FOUND

ii) Csak üres sorokat tartalmazó állomány:

Bemenet: in2.txt

Kimenet: NOT FOUND

iii) Egyetlen (feltételnek megfelelő) paragrafust tartalmazó állomány

Bemenet: in3.txt

Kimenet:

id: 2

Number of rows: 3

Number of words: 6

Ratio: 2

iv) Több (feltételnek megfelelő) paragrafust tartalmazó állomány

Bemenet: in4.txt

Kimenet:

id: 2

Number of rows: 3

Number of words: 6

Ratio: 2

b) Tétel szerint:

i) Az első elem maximális:

Bemenet: in5.txt

Kimenet:

id: 1

Number of rows: 4

Number of words: 6

Ratio: 1.5

ii) Az utolsó elem maximális:

Bemenet: in6.txt

Kimenet:

id: 2

Number of rows: 4

Number of words: 6

Ratio: 1.5

iii) Minden elem maximális:

Bemenet: in7.txt

Kimenet:

id: 1

Number of rows: 4

Number of words: 6

Ratio: 1.5

2) Egy bekezdéshez tartozó sorok szintjén (megszámlálás):

a) *Intervallum hossza szerint:*

i) *Egy sort tartalmaz:*

Bemenet: in8.txt

Kimenet: NOT FOUND

ii) *Több sort tartalmaz:*

Bemenet: in9.txt

Kimenet:

id: 2

Number of rows: 4

Number of words: 6

Ratio: 1.5

3) Egy sor szintjén (Megszámlálás):

a) *Intervallum hossza szerint:*

i) *Egy szó van:*

Bemenet: in10.txt

Kimenet:

id: 1

Number of rows: 7

Number of words: 7

Ratio: 1

ii) Több szó van:

Bemenet: in11.txt

Kimenet:

id: 2

Number of rows: 4

Number of words: 21

Ratio: 5.25