

”Programozás”  
beadandó feladat:  
9. feladat

Készítette: Bárdosi Bence  
Neptun-azonosító: VY9NJN  
E-mail: bardosi.bence@gmail.com

2017-04-18

# Tartalom

<b>1</b>	<b>Dokumentáció</b>	<b>2</b>
1.1	Feladat . . . . .	2
1.2	Specifikáció . . . . .	2
1.3	Algoritmus . . . . .	2
1.4	Implementáció . . . . .	2
1.4.1	Adattípusok megvalósítása . . . . .	2
1.4.2	Bemenő adatok formája . . . . .	2
1.4.3	Program váz . . . . .	3
1.5	Tesztelés . . . . .	4
1.5.1	A programozási tételekre épülő (szürke doboz) tesztesetek: . . . . .	4
1.5.2	A megoldó programra épülő (fehér doboz) tesztesetek: . . . . .	4

# 1 Dokumentáció

## 1.1 Feladat

Madarak életének kutatásával foglalkozó szakemberek  $n$  különböző településen  $m$  különböző madárfaj előfordulását tanulmányozzák. Egy adott időszakban megszámolták, hogy az egyes településen egy madárfajnak hány egyedével találkoztak. Volt-e olyan település, ahol mindegyik madárfaj előfordult?

## 1.2 Specifikáció

$$\mathbf{A} = (adat : \mathbb{N}^{n \times m}, l : \mathbb{L})$$

$$\mathbf{Ef} = (adat = adat')$$

$$\mathbf{Uf} = \left( Ef \wedge (l, \_) = \text{SEARCH}_{i=1}^n mind(i) \right)$$

$$\mathbf{ahol} \ mind(i) : \mathbb{N} \rightarrow \mathbb{L}$$

$$\mathbf{és} \forall i \in [1..n] : mind(i) = \forall \text{SEARCH}_{j=1}^m adat[i, j] > 0$$

## 1.3 Algoritmus

A feladatot a lineáris keresés, az alfeladatot az optimista lineáris keresés programozási tételeire vezetjük vissza.

Lin. ker.		
Tétel		Feladat
$m$	$\leftarrow$	1
$n$	$\leftarrow$	$n$
$ind$	$\leftarrow$	-
$\beta(i)$	$\leftarrow$	$mind(i)$

$l, i := hamis, 1$
$\neg l \wedge i \leq n$
$l := mind(i)$
$i := i + 1$

Opt. lin. ker.		
Tétel		Feladat
$m$	$\leftarrow$	1
$n$	$\leftarrow$	$m$
$i$	$\leftarrow$	$j$
$\beta(i)$	$\leftarrow$	$adat[i, j] > 0$

$l := mind(i)$

$l, j := true, 1$
$l \wedge j \leq m$
$l := adat[i, j] > 0$
$j := j + 1$

## 1.4 Implementáció

### 1.4.1 Adattípusok megvalósítása

A tervben szereplő mátrixot `vector<vector<int>>`-ként deklaráljuk. Mivel a vektor 0-tól indexelődik, azért a tervbeli ciklusok indextartományai a  $0..n\{1}$  és a  $0..m\{1}$  intervallumra módosulnak, ahol a  $n$ -re `t.size()` alakban,  $m$ -re pedig `t[i].size()` alakban hivatkozhatunk.

### 1.4.2 Bemenő adatok formája

Az adatokat be lehet olvasni egy szöveges állományból vagy meg lehet adni billentyűzetről. A program először megkérdezi az adatbevitel módját, majd a szöveges állományból való olvasást választva bekéri az állomány nevét. A billentyűzetről vezérelt adatbevitelt a program párbeszéd-üzemmódban irányítja, és azt megfelelő adat-ellenőrzésekkel vizsgálja. A szöveges állomány formája kötött, arról feltesszük, hogy helyesen van kitöltve, ezért ezt külön nem ellenőrizzük. Az első sor a városok és a madárfajok számát tartalmazza, szóközzel vagy tabulátor jelekkel elválasztva. Ezt követően következnek a városonként megfigyelt madárfajok egyedeinek száma kötött sorrendben. A fájl egy sorvége jel zárja.

### 1.4.3 Program váz

A program több állományból áll. A read csomag (`read.h`, `read.cpp`) felel az adatok helyes beolvasásáról és azok ellenőrzéséről. A "madar" csomag (`madar.h`, `madar.cpp`) felel feladat megoldásáért. Végül a "teszt" csomag (`catch.hpp`, `test.h`, `test.cpp`) az egységteszt megszervezéséért felel. A csomagokban található függvények/állományok feladatai az alábbi táblázatokból olvashatók.

Read csomag

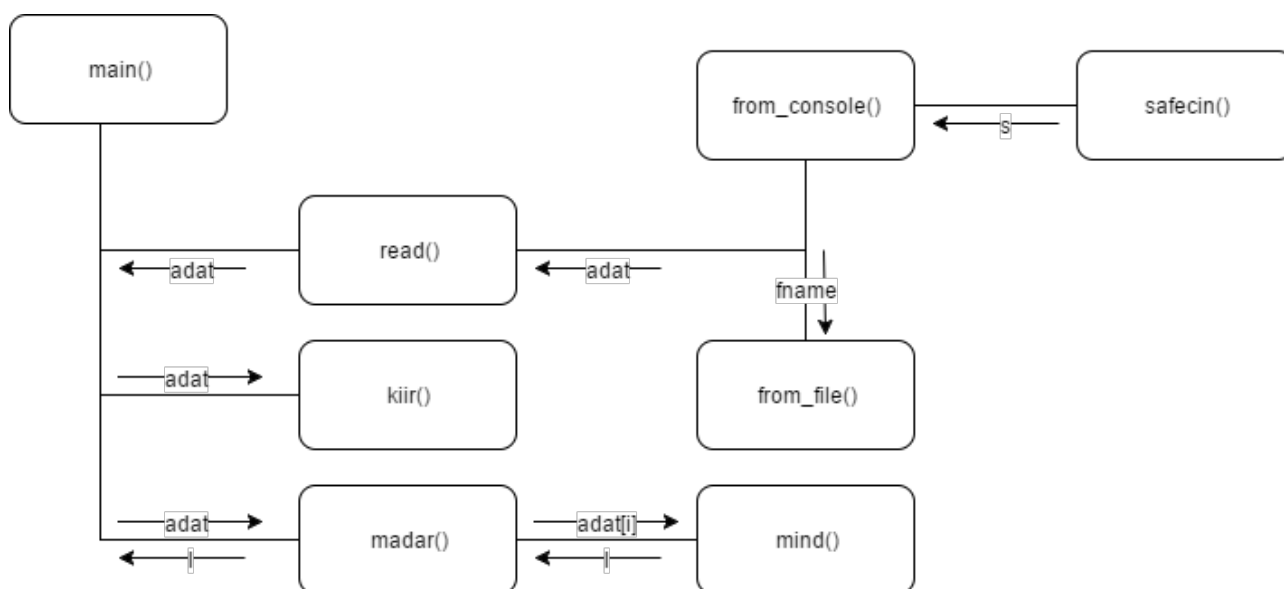
Függvény	Feladat
<code>read</code>	Megkérdezi a felhasználótól, hogy milyen módon kívánja az adatokat bevinni.
<code>from_file</code>	Megnyitja a megadott szöveges állományt, és beolvassa az adatokat.
<code>from_console</code>	Párbeszédes formában bekéri a felhasználótól az adatokat.
<code>safecin</code>	Ellenőrzi, hogy a kapott adat természetes szám-e. Hiba esetén értesíti a felhasználót.

Madar csomag

Függvény	Feladat
<code>madar</code>	Megadja, hogy az adott mátrixban létezik-e megfelelő tulajdonságú város.
<code>mind</code>	Megadja, hogy az adott tömbben minden madárfajból előfordul-e legalább 1.

Test csomag

Állomány	Feladat
<code>test.h</code>	Tartalmazza azt a flag-et, amely megszervezi, hogy programfutás, vagy egységteszt forduljon.
<code>test.cpp</code>	Egységtesztet tartalmazó állomány
<code>catch.hpp</code>	Külső könyvtár, mely megszervezi az automatikus egységtesztet.



## 1.5 Tesztelés

### 1.5.1 A programozási tételekre épülő (szürke doboz) tesztesetek:

Külső programozási tétel (lin.ker)

---

**intervallum hossza** szerint:

1. *nulla* hosszú: 0 város, 0 madár  
be11.txt - válasz: Hamis
2. *egy* város: van "0" madár  
be12.txt - válasz: Hamis
3. *egy* város: nincs "0" madár  
be13.txt - válasz: Igaz
4. *több* város:  
be14.txt - válasz: Igaz

**intervallum eleje** szerint:

Csak az első városban fordult elő az összes madárból  
be15.txt - válasz: Igaz

**intervallum vége** szerint:

Csak az utolsó városban fordult elő az összes madárból  
be16.txt - válasz: Igaz

**tételre jellemző esetek** szerint:

1. Több város, több madár mindegyik városban előfordult mind:  
be17.txt - válasz: Igaz
  2. Több város, több madár egy városban előfordult mind:  
be18.txt - válasz: Igaz
  3. Több város, több madár egy városban sem fordult elő mind:  
be19.txt - válasz: Hamis
- 

Belső programozási tétel (opt.lin.ker)

---

**intervallum hossza** szerint:

1. *nulla* hosszú: 0 madár  
be21.txt - válasz: Igaz
2. *egy* "0" madár  
be22.txt - válasz: Hamis
3. *egy* nem "0" madár  
be23.txt - válasz: Igaz
4. *több* madár:  
be24.txt - válasz: Igaz

**intervallum eleje** szerint:

Csak az első madár "0"  
be25.txt - válasz: Hamis

**intervallum vége** szerint:

Csak az utolsó madár "0"  
be26.txt - válasz: Hamis

**tételre jellemző esetek** szerint:

1. Több madár, mind nem "0":  
be27.txt - válasz: Igaz
  2. Több madár, egy "0":  
be28.txt - válasz: Hamis
  3. Több madár, több 0:  
be29.txt - válasz: Hamis
- 

### 1.5.2 A megoldó programra épülő (fehér doboz) tesztesetek:

- Hibás vagy nem létező állománynév megadása.
- Menü választás tesztelése.

- Beolvasás mindkét módozatána tesztelése.
- Hibás adatok beolvasásának tesztelése.