

Programozás példa zh

Feladatok

Minden feladatnál:

- a) Nevezd meg az alkalmazott programozási tétel(eke)t! (Elég a visszavezetési táblázat fejléceként megadni.)
- b) Írd fel a feladat specifikációját! (Az utófeltételben használd a tételek rövidített formáját! A feladatspecifikáció és programspecifikáció formáját is használhatod, de következetesen!)
- c) Írd fel a visszavezetési táblázato(ka)t!
- d) Írd fel a feladat struktogramját! (Nagyon fontos, hogy az algoritmusnak meg kell felelnie a tanított algoritmussémáknak!)

1. Palindrom

Igaz-e, hogy egy karaktertömbben elhelyezett szöveg odafelé és visszafelé olvasva is ugyanaz?

2. Legmagasabb csúcs

Egy túrázó GPS-e rögzítette a túra egy szakaszát, 100m-enként tárolta a koordináták mellett a tengerszint feletti magasságot is. Add meg a legmagasabb csúcsot! Csúcsnak azt nevezzük, amikor egy pont az előző és a következő pontnál is magasabban helyezkedik el.

3. Helységek több mint 90%-ban egyféle madárral

Az ország néhány helységében madármegfigyelést végeztünk. Mindegyikben megadtuk, hogy milyen fajú madárból hányat láttunk. Add meg azokat a helységeket, ahol a madarak több mint 90%-a egyféle madár volt!

4. Leghosszabb sziget

Európából Amerikába repülünk és közben mérjük a tengerszint feletti magasságot. 0-t mérünk, ahol tenger van, ennél nagyobb szárazföld esetén. Add meg a leghosszabb sziget kezdő és végpontját!

Megoldások

1. Palindrom

Igaz-e, hogy egy karaktertömbben elhelyezett szöveg odafelé és visszafelé olvasva is ugyanaz?

Specifikáció

Be: n : Egész, s : Tömb($1..n$: Karakter)

Ki: $palindrome$: Logikai

Ef: $n = n'$ és $s = s'$ és $n \geq 0$

Uf: Ef és $palindrome = \forall_{i=1}^{n \div 2} s[i] = s[n - i + 1]$

(Optimista) keresés (eldöntés)

$mind$	\sim	$palindrome$
$e..u$	\sim	$1..n \div 2$
$T(i)$	\sim	$s[i] = s[n - i + 1]$

Algoritmus

$i:=1$
$i \leq n \div 2$ és $s[i]=s[n-i+1]$
$i:=i+1$
$palindrome:=i > n \div 2$

2. Legmagasabb csúcs

Egy túrázó GPS-e rögzítette a túra egy szakaszát, 100m-enként tárolta a koordináták mellett a tengerszint feletti magasságot is. Add meg a legmagasabb csúcsot! Csúcsnak azt nevezzük, amikor egy pont az előző és a következő pontnál is magasabban helyezkedik el.

A feladat többféleképpen is megoldható, a lényeg, hogy a tanult tételekre vezessük vissza!

- a) *Feltételes maximumkeresés*
- b) *Kiválogatás és maximumkiválasztás*

Nézzük ez utóbbit!

Specifikáció

Be: n : Egész, mag : Tömb($1..n$: Valós)

Ki: van : Logikai, $legmagasabbcsúcs$: Valós

Ef: $n = n'$ és $mag = mag'$ és $n \geq 0$

Uf: Ef és

$csúcsok$: Tömb($1..n$: Egész) és $(db, csúcsok) = Kiválogat^{n-1}_{mag[i-1] < mag[i] > mag[i+1]} i$ és $i=2$

$van = db > 0$ és

$van \rightarrow (legmagasabbcsúcs) = Max_{i=1}^{db} mag[csúcsok[i]]$

Kiválogatás

y	\sim	$csúcsok$
$e..u$	\sim	$2..n - 1$
$T(i)$	\sim	$mag[i - 1] < mag[i] > mag[i + 1]$
$f(i)$	\sim	i

Maximumkiválasztás

$maxért$	\sim	$legmagasabbcsúcs$
$e..u$	\sim	$1..db$
$f(i)$	\sim	$mag[csúcsok[i]]$

Algoritmus

db:=0		
i=2..n-1		
<div> <div>mag[i-1]<mag[i]>mag[i+1]</div> <div> <div>true</div> <div>false</div> </div> </div>		
db:=db+1	-	
csúcsok[db]:=i		
van:=db>0		
<div> <div>van</div> <div> <div>true</div> <div>false</div> </div> </div>		
maxért:=mag[csúcsok[1]]; maxind:=1	-	
i=2..db		
<div> <div>mag[csúcsok[i]]>maxért</div> <div> <div>true</div> <div>false</div> </div> </div>		
maxért:=mag[csúcsok[i]]	-	
maxind:=i		

3. Helységek több mint 90%-ban egyféle madárral

Az ország néhány helységében madármegfigyelést végeztünk. Mindegyikben megadtuk, hogy milyen fajú madárból hányat láttunk. Add meg azokat a helységeket, ahol a madarak több mint 90%-a egyféle madár volt!

Specifikáció

Be: n, m : Egész, $madarak$: Tömb($1..n, 1..m$: Egész)

Ki: db : Egész, $hely$: Tömb($1..db$: Egész)

Ef: $n = n'$ és $m = m'$ és $madarak = madarak'$ és $n \geq 0$ és $m \geq 0$

Uf: Ef és $(db, hely) = Kiválogat^n$ $\frac{\text{maximum}(sor)}{\text{összeg}(sor)} > 0.9$ sor és

$maximum$: Egész \rightarrow Egész és

$maximum(sor) = \text{Max}_{oszlop=1}^m madarak[sor, oszlop]$ és

$összeg$: Egész \rightarrow Egész és

$összeg(sor) = \sum_{oszlop=1}^m madarak[sor, oszlop]$

Kiválogatás

$y \sim hely$
 $i \sim sor$
 $e..u \sim 1..n$
 $T(i) \sim \frac{\text{maximum}(sor)}{\text{összeg}(sor)} > 0.9$
 $f(i) \sim i$

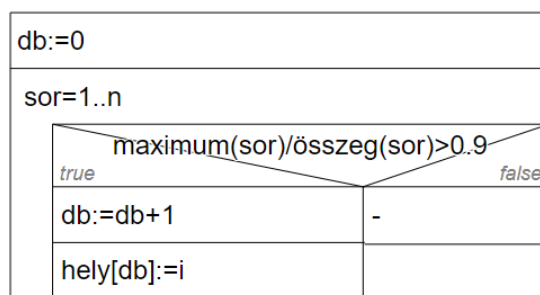
Maximumkiválasztás

$i \sim oszlop$
 $e..u \sim 1..m$
 $f(i) \sim madarak[sor, oszlop]$

Összegezés

$i \sim oszlop$
 $e..u \sim 1..m$
 $f(i) \sim madarak[sor, oszlop]$

Algoritmus



maximum(sor: Egész): Egész

maxért:=madarak[sor,1]; maxind:=1	
oszlop=2..m	
madarak[sor,oszlop]>maxért	
true	false
maxért:=madarak[sor,oszlop]	-
maxind:=oszlop	
maximum:=maxért	

összeg(sor:Egész): Logikai

s:=0
oszlop=1..m
s:=s+madarak[sor,oszlop]
összeg:=s

4. Leghosszabb sziget

Európából Amerikába repülünk és közben mérjük a tengerszint feletti magasságot. 0-t mérünk, ahol tenger van, ennél nagyobb szárazföld esetén. Add meg a leghosszabb sziget kezdő és végpontját!

Specifikáció

Be: n : Egész, mag : Tömb($1..n$: Valós)

Ki: van : Logikai, k, v : Egész

Ef: $n = n'$ és $mag = mag'$ és $n \geq 0$

Uf: Ef és db : Egész és $szigetek$: Tömb($1..db$: Sziget) és

$Sziget = Rekord(kezdet, vég: Egész)$ és

$(db, szigetek) = Kiválogat^n_{szk(i) \text{ és } vanvég(i)} Sziget(i, holvég(i))$ és

$szk: Egész \rightarrow Logikai$ és $szk(i) = i > 1$ és $mag[i - 1] = 0$ és $mag[i] > 0$ és

$szv: Egész \rightarrow Logikai$ és $szv(i) = i < n$ és $mag[i + 1] = 0$ és $mag[i] > 0$ és

$vanvég: Egész \rightarrow Logikai$ és $vanvég(i) = \exists_{j=i}^n szv(j)$ és

$holvég: Egész \rightarrow Egész$ és $holvég(i) = Kiválaszt_{j \geq i} szv(j)$ és

$(van, maxind) = Max_{igaz}_{i=1}^{db} szigetek[i].vég - szigetek[i].kezdet$ és

$van \rightarrow (k, v) = szigetek[maxind]$

Kiválogatás

y	\sim	$szigetek$
$e..u$	\sim	$1..n$
$T(i)$	\sim	$szk(i)$ és $vanvég(i)$
$f(i)$	\sim	$Sziget(i, holvég(i))$

Keresés (eldöntés)

i	\sim	j
$e..u$	\sim	$i..n$
$T(i)$	\sim	$szv(j)$

Kiválasztás

i	\sim	j
$e..u$	\sim	$i..n$
$T(i)$	\sim	$szv(j)$

Feltételes maximumkeresés

$e..u$	\sim	$1..db$
$T(i)$	\sim	$igaz$
$f(i)$	\sim	$szigetek[i].vég - szigetek[i].kezdet$

Algoritmus

db:=0			
i=1..n			
szk(i) és vanvég(i)			
db:=db+1		-	
szigetek[db]:=Sziget(i, holvég(i))			
van:=hamis			
i=1..db			
nem igaz	van és igaz		nem van és igaz
-	szigetek[i].vég-szigetek[i].kezdet > maxért		van:=igaz
	maxért:= szigetek[i].vég-szigetek[i].kezdet		- maxért:= szigetek[i].vég-szigetek[i].kezdet
	maxind:=i		
van			
(k,v):=szigetek[maxind]		-	

vanvég(i:Egész): Logikai

j:=i
j≤n és nem szv(j)
j:=j+1
van:=j≤n
vanvég:=van

holvég(i:Egész): Logikai

j:=i
nem szv(j)
j:=j+1
ind:=j
holvég:=ind