

7. táblás gyakorlat feladatai

1. Egy karakterekből álló szekvenciális inputfájlban melyik az a leghosszabb szó, amelyik tartalmazza a 'w' betűt? (Egy szót szóközök vagy a fájl vége határol.)

Specifikáció:

$A = (x:infile(\mathbb{K}), l:\mathbb{L}, szó:\mathbb{S})$
 $Ef = (x = x_0)$

Ötlet:

Soroljuk fel a szavakat, azzal az információval együtt, hogy van-e bennük 'w' betű.

Új Specifikáció:

$A = (t:enor(Szó), l:\mathbb{L}, szó:\mathbb{S})$
 $Szó = rec(szó:\mathbb{S}, w:\mathbb{L})$
 $Ef = (t = t_0)$
 $Uf = ((l, max, elem) = \underset{e.w}{MAX_{e \in t_0}} |e.szó|) \wedge (l \rightarrow szó=elem.szó)$

Feltételes maximum keresés

$f(e) \sim |e.szó|$
 $felt(e) \sim e.w$
 $H, > \sim \mathbb{N}, >$

Algoritmus:

l := hamis; t.first()			
¬t.end()			
¬ t.current().w	t.current().w ∧ l		t.current().w ∧ ¬l
—	t.current().szó > max		l, max, szó := igaz, t.current().szó , t.current().szó
	max, szó := t.current().szó , t.current().szó		
t.next()			

Felsoroló:

$t:enor(Szó)$ $Szó = rec(szó:\mathbb{S}, w:\mathbb{L})$

Szó*	first()	next()	current() : Szó	end() : \mathbb{L}
$x:infile(\mathbb{K})$ $dx:\mathbb{K}$ $sx:Status$ $akt:Szó$ $vége:\mathbb{L}$	$sx, dx, x:read$ $next()$	lásd külön	return akt	return vége

next() művelet

$A = (x:infile(\mathbb{K}), dx:\mathbb{K}, sx:Status, akt:Szó, vége:\mathbb{L})$

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

$Uf = ((dx'', (sx'', dx'', x'')) = SELECT_{dx \in (dx', x')} (sx=abnorm \vee dx \neq ''))$

$\wedge vége=(sx''=abnorm)$

$dx \neq ''$

$dx \neq ''$

$\wedge (\neg vége \rightarrow (akt.szó, (sx, dx, x)) = \bigoplus_{dx \in (dx'', x'')} <dx> \wedge (akt.w, (sx, dx, x)) = \bigvee_{dx \in (dx'', x'')} dx='w'))$

Megj: A „van 'w' karakter egy szóban” részfeladatot lineáris keresés helyett összevagyolással számoljuk ki, mert így is, úgy is végig kell menni a teljes aktuális szón (lásd a szó betűinek összefűzését), amellyel az összevagyolást össze lehet vonni közös felsorolásba (ciklusba).

Kiválasztás

t:enor(E) ~ x:infile(\mathbb{K}) (sx,dx,x:read)
 first() nélkül
 felt(e) ~ sx=abnorm \vee dx \neq ' '

Két összegzés (összefűzés és összevagyolás)

t:enor(E) ~ x:infile(\mathbb{K}) (sx,dx,x:read)
 first() nélkül, felt: dx \neq ' '
 f(e) ~ (<dx>, dx='w')
 s ~ (akt.szó, akt.w)
 H, +, 0 ~ (\mathbb{K}^* , \mathbb{L}), (\oplus , \vee), (<>, hamis)

sx=norm \wedge dx=' '	
sx,dx,x:read	
vége := sx=abnorm	
\neg vége	
akt.szó, akt.w := <>, hamis	–
sx=norm \wedge dx \neq ' '	
akt.szó, akt.w := akt.szó \oplus <dx>, akt.w \vee (dx='w')	
sx,dx,x:read	

2. Egy szekvenciális inputfájlban a vadászok eredményeit (vadász neve, vadászat dátuma, zsákmány fajtája, zsákmány súlya) tároljuk a vadászok neve, azon belül a vadászat dátuma szerint rendezetten. Igaz-e, hogy minden vadász valamelyik vadászatán lőtt medvét?

Specifikáció:

$A = (x:\text{infile}(\text{Trófea}), l:\mathbb{L})$
 $\text{Trófea} =$
 $\text{rec}(\text{név}:\mathbb{S}, \text{dátum}:\mathbb{S}, \text{fajta}:\mathbb{S}, \text{súly}:\mathbb{N})$
 $Ef = (x=x_0 \wedge x \nearrow (\text{név}, \text{dátum}))$

Ötlet:

Soroljunk fel annyi logikai értéket ahány vadász van: egy logikai érték akkor igaz, ha a vadász lőtt medvét.

Optimista lineáris keresés

$\text{felt}(e) \sim e$

Új specifikáció:

$A = (t:\text{enor}(\mathbb{L}), l:\mathbb{L})$
 $Ef = (t=t_0)$
 $Uf = (l = \forall \text{SEARCH}_{e \in t_0} e)$

Algoritmus:

$l := \text{igaz}$ $t.\text{first}()$
$l \wedge \neg t.\text{end}()$
$l := t.\text{current}()$
$t.\text{next}()$

Felsoroló:

$t:\text{enor}(\mathbb{L})$

\mathbb{L}^*	$\text{first}()$	$\text{next}()$	$\text{current}() : \mathbb{L}$	$\text{end}() : \mathbb{L}$
$x : \text{infile}(\text{Trófea})$ $dx : \text{Trófea}$ $sx : \text{Status}$ $akt : \mathbb{L}$ $vége : \mathbb{L}$	$sx, dx, x:\text{read}$ $\text{next}()$	lásd külön	return akt	return vége

$\text{Trófea} = \text{rec}(\text{név}:\mathbb{S}, \text{dátum}:\mathbb{S}, \text{fajta}:\mathbb{S}, \text{súly}:\mathbb{N})$

next() művelet

$A = (x:\text{infile}(\text{Trófea}), dx:\text{Trófea}, sx:\text{Status}, akt:\mathbb{L}, vége:\mathbb{L})$

$Ef = (x = x' \wedge dx = dx' \wedge sx = sx')$ $dx.\text{név} = dx'.\text{név}$

$Uf = (vége = (sx' = \text{abnorm}) \wedge (\neg vége \rightarrow (akt, (sx, dx, x)) = \bigvee_{dx \in (dx', x')} dx.\text{fajta} = \text{"medve"}))$

Összegzés (összevagyolás)

$t:\text{enor}(E) \sim x:\text{infile}(\text{Trófea}) (sx, dx, x:\text{read})$
 $\text{first}()$ nélkül,
 $\text{felt: } dx.\text{név} = dx'.\text{név}$
 $f(e) \sim dx.\text{fajta} = \text{"medve"}$
 $s \sim akt$
 $H, +, 0 \sim \mathbb{L}, \vee, <>$

$vége := sx = \text{abnorm}$	
$\neg vége$	
$akt := \text{hamis}$	–
$név := dx.\text{név}$	
$sx = \text{norm} \wedge dx.\text{név} = név$	
$akt := akt \vee (dx.\text{fajta} = \text{"medve"})$	
$sx, dx, x:\text{read}$	

3. Egy szekvenciális inputfájlban a vadászok eredményeit (vadász neve, vadászat dátuma, zsákmány fajtája, zsákmány súlya) tároljuk a vadászok neve, azon belül a vadászat dátuma szerint rendezetten. Hány olyan vadász volt, aki minden vadászatán lőtt nyulat, és a zsákmányainak összsúlya meghaladta a 250 kilogrammot.

Specifikáció:

$A = (x:\text{infile}(\text{Trófea}), c: \mathbb{N})$
 $\text{Trófea} =$
 $\text{rec}(\text{név}:\mathbb{S}, \text{dátum}:\mathbb{S}, \text{fajta}:\mathbb{S}, \text{súly}:\mathbb{N})$
 $Ef = (x=x_0 \wedge x \nearrow (\text{név}, \text{dátum}))$

Ötlet:

Soroljuk fel a vadászok neveit azzal az információval együtt, hogy lőttek-e minden vadászatukon nyulat, és mennyi volt a zsákmányaiknak az összsúlya.

Új specifikáció:

$A = (t:\text{enor}(\text{Vadász}), c: \mathbb{N})$
 $\text{Vadász} = \text{rec}(\text{név}:\mathbb{S}, \text{nyúl}:\mathbb{L}, \text{összsúly}:\mathbb{N})$
 $Ef = (t=t_0)$
 $Uf = (c = \sum_{e \in t_0} 1)$
 $e.\text{nyúl} \wedge e.\text{összsúly} > 250$

Algoritmus:

$c := 0$	
$t.\text{first}()$	
$\neg t.\text{end}()$	
$t.\text{current}().\text{nyúl} \wedge$ $t.\text{current}().\text{összsúly} > 250$	
$c := c + 1$	–
$t.\text{next}()$	

Számlálás

$\text{felt}(e) \quad \sim e.\text{nyúl} \wedge e.\text{összsúly} > 250$

Felsoroló:

Ötlet: A $t:\text{enor}(\text{Vadász})$ felsoroló megvalósításához tegyük fel, hogy rendelkezünk olyan $y:\text{enor}(\text{Összesítés})$ felsorolóval, amely a vadászoknak az egyes vadászatokon elért összesített eredményeit sorolja fel a vadász neve és a vadászat dátuma szerint növekedően rendezetten: lőtt-e egy adott vadászatán nyulat, és mennyi volt ott a zsákmányai súlyának összege.

$t:\text{enor}(\text{Vadász})$ $\text{Vadász} = \text{rec}(\text{név}:\mathbb{S}, \text{nyúl}:\mathbb{L}, \text{összsúly}:\mathbb{N})$

Vadász*	first()	next()	current() : Vadász	end() : \mathbb{L}
$y : \text{enor}(\text{Összesítés})$ $\text{akt} : \text{Vadász}$ $\text{vége} : \mathbb{L}$	$y.\text{first}()$ $\text{next}()$	lásd külön	return akt	return vége

$\text{Összesítés} = \text{rec}(\text{név}:\mathbb{S}, \text{dátum}:\mathbb{S}, \text{lőtt}:\mathbb{L}, \text{össz}:\mathbb{N})$

next() művelet

$A = (y:\text{enor}(\text{Összesítés}), \text{akt}:\text{Vadász}, \text{vége}:\mathbb{L})$

$Ef = (y = y' \wedge y \nearrow (\text{név}, \text{dátum}))$

$Uf = (\text{vége} = y'.\text{end}() \wedge (\neg \text{vége} \rightarrow \text{akt}.\text{név} = y'.\text{current}().\text{név}))$

$e.\text{név} = \text{akt}.\text{név}$

$\wedge (\text{akt}.\text{nyúl}, y) = \bigwedge_{e \in (y'.\text{current}(), y')} e.\text{lőtt}$

$e.\text{név} = \text{akt}.\text{név}$

$\wedge (\text{akt}.\text{összsúly}, y) = \sum_{e \in (y'.\text{current}(), y')} e.\text{össz} \quad)$

Két összegzés (éselés és összeadás)

közös felsorolóval

t:enor(E) ~ y:enor(Összesítés)
 first() nélkül
 felt: y.current().név=akt.név
 f(e) ~ y.current().lőtt
 s ~ akt.nyúl
 H, +, 0 ~ \mathbb{L} , \wedge , igaz
 f(e) ~ y.current().össz
 s ~ akt.összsúly
 H, +, 0 ~ \mathbb{N} , +, 0

vége := y.end()	
¬vége	
akt := (y.current().név, igaz, 0)	–
¬y.end() ∧ y.current().név=akt.név	
akt.nyúl := akt.nyúl ∧ y.current().lőtt	
akt.összsúly := akt.összsúly + y.current().össz	
y.next()	

Felsoroló:

y:enor(Összesítés) Összesítés = rec(név:S, dátum:S, lőtt:L, össz:N)

Összesítés*	first()	next()	current() : Összesítés	end() : L
x : infile(Trófea) dx : \mathbb{K} sx : Status akt : Összesítés vége : L	sx,dx,x:read next()	lásd külön	return akt	return vége

Trófea = rec(név:S, dátum:S, fajta:S, súly:N)

next() művelet

A = (x:infile(Trófea), dx:Trófea, sx:Status, akt:Összesítés, vége:L)

Ef = (x = x' \wedge x' \nearrow (név,dátum) \wedge dx = dx' \wedge sx = sx')

Uf = (vége = (sx'=abnorm) \wedge (\neg vége \rightarrow akt.név=dx'.név \wedge akt.dátum=dx'.dátum \wedge

dx,név=akt.név \wedge dx,dátum=akt.dátum

\wedge (akt.lőtt, (sx,dx,x)) = $\bigvee_{dx \in (dx',x')} dx.fajta = \text{"nyúl"} \wedge$

dx,név=akt.név \wedge dx,dátum=akt.dátum

\wedge (akt.össz, (sx,dx,x)) = $\sum_{dx \in (dx',x')} dx.súly$))

Két összegzés (vagylás és összeadás)

közös felsorolóval

t:enor(E) ~ x:infile(Eredmény)
 sx,dx,x:read, first() nélkül,
 felt: dx.név=akt.név
 \wedge dx.dátum=akt.dátum
 f(e) ~ dx.fajta="nyúl"
 s ~ akt.lőtt
 H, +, 0 ~ \mathbb{L} , \vee , hamis
 f(e) ~ dx.súly
 s ~ akt.össz
 H, +, 0 ~ \mathbb{N} , +, 0

vége := sx=abnorm	
¬vége	
akt := (dx.név, dx.dátum, hamis, 0)	–
sx=norm ∧ dx.név=akt.név ∧ dx.dátum=akt.dátum	
akt.lőtt := akt.lőtt ∨ (dx.fajta="nyúl")	
akt.össz := akt.össz + dx.súly	
sx,dx,x:read	