

5. táblás gyakorlat feladatai

1. Egy szekvenciális inputfájlban kaktuszfajtákról tárolunk adatokat: név, őshaza, virágszín, méret.
Kaktusz=rec(név:\$, szín:\$, ős:\$, méret:\$)

- b) Van-e piros virágú kaktusz?

Specifikáció:

$A = (x:\text{infile}(\text{Kaktusz}), l:\mathbb{L})$

$Ef = (x=x_0)$

$Uf = (l = \text{SEARCH}_{e \in x_0} e.\text{szín} = \text{"piros"})$

Linker

$t:\text{enor}(E) \sim x:\text{infile}(\text{Kaktusz}) \ (st, e, x:\text{read})$

$\text{felt}(e) \sim e.\text{szín} = \text{"piros"}$

$(? l = \forall \text{SEARCH}_{e \in x_0} e.\text{szín} = \text{"piros"})$

Algoritmus:

l := hamis	
¬l ∧ e in x	
e.szín="piros"	
l := igaz	—

l := hamis	
First()	
¬l ∧ ¬End()	
Current().szín="piros"	
l := igaz	Next()

l := hamis	
st, e, x : read	
¬l ∧ st=norm	
e.szín="piros"	
l := igaz	st, e, x : read

- c) Melyik a legnagyobb piros virágú kaktusz neve?

Specifikáció:

$A = (x:\text{infile}(\text{Kaktusz}), l:\mathbb{L}, \text{név}:\mathbb{S})$

$Ef = (x=x_0)$

$Uf = ((l, \text{max}, \text{elem}) = \text{MAX}_{e \in x_0} e.\text{méret} \wedge l \rightarrow \text{név} = \text{elem}.\text{név}) \quad \text{max}:\mathbb{Z}, \text{elem}:\text{Kaktusz}$
 $e.\text{szín} = \text{"piros"}$

Felt. max. ker.

$t:\text{enor}(E) \sim x:\text{infile}(\text{Kaktusz}) \ (st, e, x:\text{read})$

$f(e) \sim e.\text{méret}$

$\text{felt}(e) \sim e.\text{szín} = \text{"piros"}$

$H, > \sim \mathbb{N}, >$

Algoritmus:

l := hamis					e:Kaktusz st:Status
st, e, x : read					
st=norm					
e.szín≠"piros"	e.szín="piros" ∧ l		e.szín="piros" ∧ ¬l		
	e.méret > max		l, max, név :=		
	max, név := e.méret, e.név	—	igaz, e.méret, e.név		
st, e, x : read					

Megj: A specifikáció szerint még egy elágazás is kellene a ciklus után, de ezt beépíthetjük a feltételes maximum keresésbe, sőt a legkisebb kaktusz tárolására az elem változóra nincs is szükségünk, helyett elég csak a név változót használni.

4.kvíz [maximum kiválasztás hibás algoritmusa]

2. Egy horgászversenyen a horgászok eredményét egy szekvenciális inputfájlban rögzítették. A fájl egy eleme egy horgász nevét és a halfogásainak sorozatát tartalmazza. Egy fogás egy időpontból, a kifogott hal fajtájának nevéből, a hal súlyából (kg) és hosszából (m) áll. Keressünk olyan horgászt, aki az 50 cm-esnél hosszabb pontyokból legalább 10 kilogramnyt fogott.

Specifikáció:

$A = (f:\text{infile}(\text{Horgász}), l:\mathbb{L}, \text{név}:\mathbb{S})$

$\text{Horgász} = \text{rec}(\text{név}:\mathbb{S}, \text{fogás}:\text{Fogás}^*)$

$\text{Fogás} = \text{rec}(\text{idő}:\mathbb{S}, \text{hal}:\mathbb{S}, \text{súly}:\mathbb{R}, \text{hossz}:\mathbb{R})$

$Ef = (f=f_0)$

$Uf = ((l, \text{elem}) = \text{SEARCH}_{h \in f_0} \text{pontysúly}(h.\text{fogás}) \geq 10.0 \wedge l \rightarrow (\text{név} = \text{elem.név}))$

ahol $\text{pontysúly}(h.\text{fogás}) = \sum_{e \in h.\text{fogás}} e.\text{súly}$
 $e.\text{hal} = \text{"ponty"} \wedge e.\text{hossz} \geq 0.5$

Lineáris keresés

$t:\text{enor}(E) \sim f:\text{infile}(\text{Horgász}) \ (st, h, f:\text{read})$

$\text{felt}(e) \sim \text{pontysúly}(h.\text{fogás}) \geq 10.0$

ahol $\text{pontysúly} : \text{Fogás}^* \rightarrow \mathbb{R}$

Algoritmus:

l := hamis		st:Status h:Horgász
st, h, f : read		
¬l ∧ st = norm		
pontysúly(h.fogás) ≥ 10.0		
l, név := igaz, h.név	st, h, f : read	

Részfeladat: $s := \text{pontysúly}(x)$

$A = (x:\text{Fogás}^*, s:\mathbb{R})$

$\text{Fogás} = \text{rec}(\text{idő}:\mathbb{S}, \text{hal}:\mathbb{S}, \text{súly}:\mathbb{R}, \text{hossz}:\mathbb{R})$

$Ef = (x=x_0)$

$Uf = (x=x_0 \wedge s = \sum_{e \in x_0} e.\text{súly})$

$e.\text{hal} = \text{"ponty"} \wedge e.\text{hossz} \geq 0.5$

Összegzés (feltételes összegzés)

$t:\text{enor}(E) \sim e \text{ in } x$

$f(e) \sim e.\text{súly}$

ha $e.\text{hal} = \text{"ponty"}$

$\wedge e.\text{hossz} \geq 0.5$

$H, +, 0 \sim \mathbb{R}, +, 0$

Algoritmus:

s := 0.0		e:Fogás
e in x		
e.hal="ponty" ∧ e.hossz≥0.5		
s := s + e.súly	—	

Megjegyzés:

Az összegzés lehetne a fájlból történő olvasás része, azaz az olvasás a horgász nevének kívül csak ezt az összeget állítaná elő, a teljes fogás listát nem.

3. Számoljuk ki egy számítástechnikai szaküzlet napi bevételét az aznapi forgalom alapján. A forgalmat a kiadott számlák mutatják, amelyeket egy szöveges állományban (szekvenciális inputfájl) rögzítettek. Az állomány minden sora egy-egy számla adatait tartalmazza: a vásárló nevét és az általa vásárolt termékek (cikkszám és ár párok) sorozatát.

Specifikáció:

$A = (f:infile(Számla), bevét:\mathbb{N})$ Számla=rec(név:\$, lista:Áru*) Áru = rec(cikkszám:\$, ár:\$)

$Ef = (f=f_0)$

$Uf = (bevét = \sum_{sz \in f_0} \text{össz}(sz.lista))$ ahol $\text{össz}(sz.lista) = \sum_{e \in sz.lista} e. \text{ár}$

Összegzés

$t:enor(E) \sim f:infile(Számla) (st,sz,f:read)$

$f(e) \sim \text{össz}(sz.lista)$

ahol $\text{össz} : \text{Áru}^* \rightarrow \mathbb{N}$

$s \sim bevét$

$H, +, 0 \sim \mathbb{N}, +, 0$

Algoritmus:

bevét := 0	st:Status sz:Számla
st, sz, f : read	
st = norm	
bevét := bevét + össz(sz.lista)	
st, sz, f : read	

Részfeladat:

sum := össz(x)

$A = (x : \text{Áru}^*, \text{sum} : \mathbb{N})$

$Ef = (x=x_0)$

$Uf = (\text{sum} = \sum_{p \in x_0} p. \text{ár})$

Összegzés

$t:enor(E) \sim p \text{ in } x$

$f(e) \sim p. \text{ár}$

$H, +, 0 \sim \mathbb{N}, +, 0$

Algoritmus:

sum := 0	p : Áru
p in x	
sum := sum+ p.ár	

Megjegyzés:

1. Az összegzés lehet egy számla (Számla típusú objektum) egy metódusa, és erre a számlára egy másik metódusával lehetne felvenni az fájlból történő olvasás során egy új tételt (árut).

A számla összege lehet a számla (Számla típusú objektum) része (adattagja), amelyet módosít, ha a számlához egy új tételt (árut) adunk hozzá.