Térinformatikai algoritmusok

Elemi algoritmusok

Cserép Máté

Analóg programozásnak nevezzük azt, amikor egy feladat megoldásához egy már ismert és megoldott feladat megoldását használjuk fel. Általában nem pontosan ugyanazt a feladatot oldottuk meg korábban, hanem egy hasonlót, azonban a két feladat fontosabb részeit meg tudjuk feleltetni egymásnak. Az analóg programozási módszerek célja a feladatok hasonló részei közötti megfeleltetések alkalmazása a megoldó programokra.

Fontos észrevenni, hogy ha csak a megoldó programot másoljuk le az új feladatra alkalmazott formában, a két feladat közötti különbséget nehéz átvezetni a programokra, könnyen tévedhetünk. Ezért gyakran nem konkrét feladatok megoldásait használjuk fel, hanem *sémákat*, vagyis olyan általános feladat-megoldás párokat, ahol bizonyos elemek paraméterként vannak megadva, és az egyes konkrét esetek határozzák meg, hogy mit kell behelyettesíteni. Ilyen sémák lehetnek a nevezetes algoritmusok, típusok és adatszerkezetek, valamint magasabb absztrakciós szinten a tervminták, elemzési minták és keretrendszerek.

Ebben a fejezetben olyan nevezetes elemi algoritmusok kerülnek bemutatásra, amelyeknek közös vonása lesz, hogy elemek egy megszámlálható, egész $[m..n]$ intervalluma felett kerül értelmezésre.[1] Az elemek értékkészlete a H halmaz.

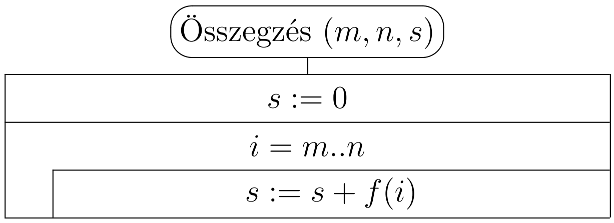
1. Összegzés

Adott $f : [m..n] \to H$ függvény. A H halmaz elemein értelmezett egy asszociatív, baloldali nulla elemmel rendelkező művelet, amelyet nevezzünk összeadásnak és jelölje a $+$ operátor szimbólum. Feladatunk, hogy az intervallumon összegezzük az f függvény felvett értékeit. Formálisan:

$$s = \sum_{i=m}^{n} f(i)$$

[1] Az üres intervallum jelölése: $[n + 1..n]$

Algoritmus:

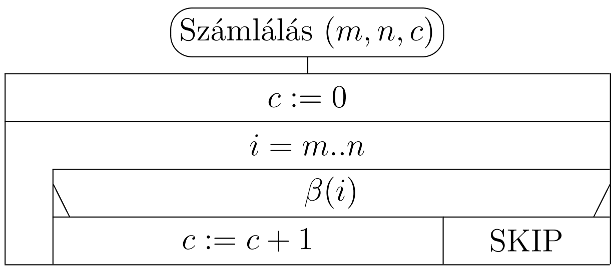


2. Számlálás

Adott $\beta : [m..n] \to \mathbb{L}$ feltétel. Határozzuk meg, hogy hány helyen teljesül az intervallumon a feltétel, azaz hányszor veszi fel az igaz értéket! Formálisan:

$$s = \sum_{\substack{i=m \\ \beta(i)}}^{n} 1$$

Algoritmus:

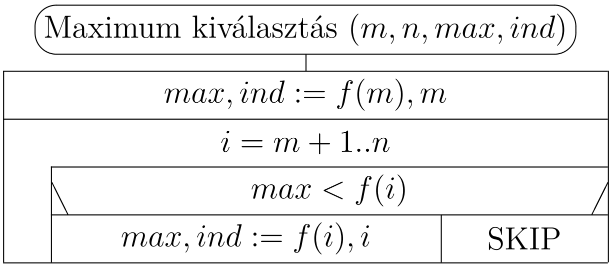


3. Maximum kiválasztás

Adott $f : [m..n] \to H$ függvény, $m \leq n$. A H halmaz elemein értelmezett egy teljes rendezési reláció (reflexív, antiszimmetrikus, tranzitív és bármely két elem összehasonlítható), amelynek jele $\leq$, erős változatának $<$. Feladatunk a függvény legnagyobb értékének meghatározása és adjuk meg az egyik olyan intervallumbeli elemet is, ahol a függvény ezt az értéket felveszi! Formálisan:

$$max = f(ind) \wedge \forall i \in [m..n] : f(i) \leq f(ind)$$

Algoritmus:

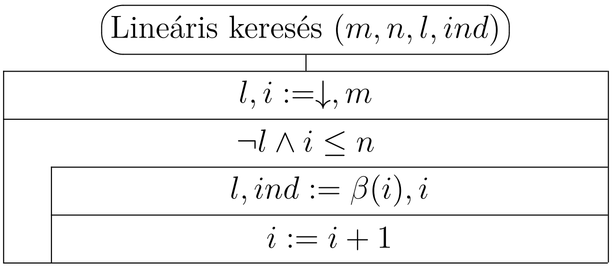


4. Lineáris keresés

Adott $\beta : [m..n] \to \mathbb{L}$ feltétel. Határozzuk meg az intervallum első olyan elemét, amelyre teljesül a feltétel, ha van egyáltalán ilyen. Formálisan:

$$l = (\exists i \in [m..n] : \beta(i))$$
$$l \to (ind \in [m..n] \land \beta(ind) \land \forall i \in [m..ind-1] : \neg\beta(i))$$

Algoritmus:



5. Algoritmusok általánosítása

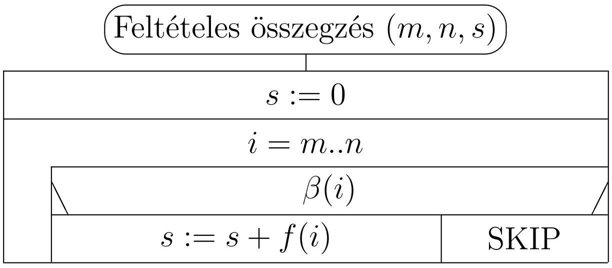
Az összegzés és a maximum kiválasztás algoritmusa tovább generalizálható, amennyiben egy $\beta : [m..n] \to \mathbb{L}$ feltétellel megszorítjuk az összeadandó, illetőleg a kiválasztandó elemek halmazát.

5.1. Feltételes összegzés

Adott $f : [m..n] \to H$ függvény és $\beta : [m..n] \to \mathbb{L}$ feltétel. A H halmaz elemein értelmezett egy asszociatív, baloldali nulla elemmel rendelkező művelet, amelyet nevezzünk összeadásnak és jelölje a $+$ operátor szimbólum. Feladatunk, hogy az intervallum azon helyein, ahol a β feltétel teljesül, összegezzük az f függvény felvett értékeit. Formálisan:

$$s = \sum_{\substack{i=m \\ \beta(i)}}^{n} f(i)$$

Algoritmus:

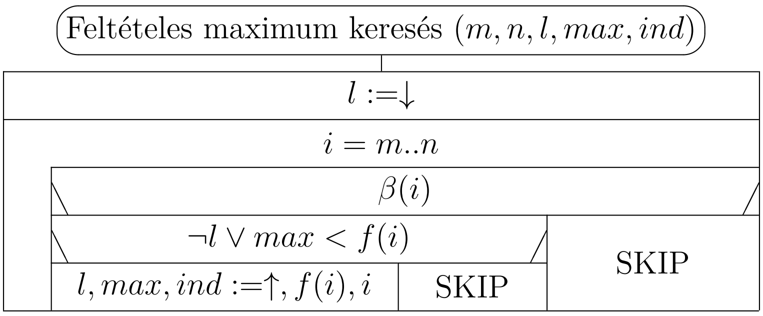


5.2. Feltételes maximum keresés

Adott $f : [m..n] \to H$ függvény és $\beta : [m..n] \to \mathbb{L}$ feltétel. A H halmaz elemein értelmezett egy teljes rendezési reláció (reflexív, antiszimmetrikus, tranzitív és bármely két elem összehasonlítható), amelynek jele $\leq$, erős változatának $<$. Határozzuk meg a függvény legnagyobb olyan értékét, amely teljesíti a β feltétel! Adjuk meg az egyik olyan intervallumbeli elemet is, ahol a függvény ezt az értéket felveszi, ha van egyáltalán ilyen. Formálisan:

$$l = (\exists i \in [m..n] : \beta(i))$$
$$l \to (\beta(ind) \wedge max = f(ind) \wedge \forall i \in [m..n] : \beta(i) \to f(i) \leq f(ind))$$

Algoritmus:

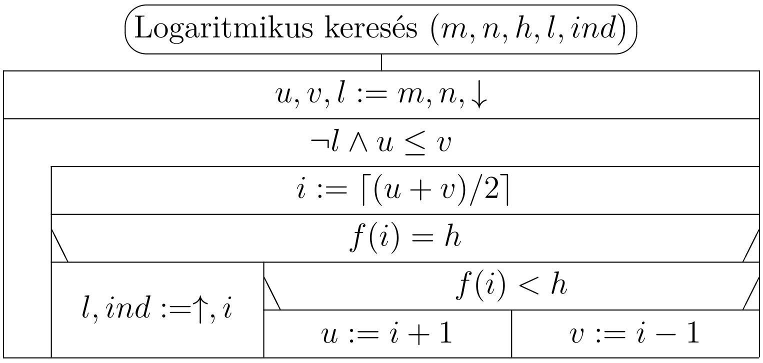


6. Bináris / logaritmikus keresés

Adott $f : [m..n] \to H$ *monoton növekvő* függvény. A H halmaz elemein értelmezett egy teljes rendezési reláció (reflexív, antiszimmetrikus, tranzitív és bármely két elem összehasonlítható), amelynek jele $\leq$, erős változatának $<$. Döntsük el, hogy az f függvény felvesz-e egy megadott $h \in H$ értéket, és ha igen, adjuk meg az egyik olyan intervallumbeli elemet is, ahol a függvény ezt az értéket felveszi. Formálisan:

$$l = (\exists i \in [m..n] : f(i) = h) \wedge l \to f(ind) = h$$

Algoritmus:


Logaritmikus keresés (m, n, h, l, ind)
$u, v, l := m, n, \downarrow$
$\neg l \wedge u \leq v$
$i := \lceil (u + v)/2 \rceil$
$f(i) = h$
$l, ind := \uparrow, i$
$f(i) < h$
$u := i + 1$
$v := i - 1$