

Algoritmusok és adatszerkezetek I.

első zárthelyi

2007. 10. 16.

1. Igaz-e, hogy $3n^3 + 2n^2 + \sqrt{n} + \frac{8}{n} = \Theta(n^3)$? (Bizonyítsd be vagy cáfold az állítást!)

Mo: Igaz, ui. $\lim_{n \rightarrow \infty} \frac{(3n^3 + 2n^2 + \sqrt{n} + \frac{8}{n})}{n^3} = 3 > 0$ konstans.

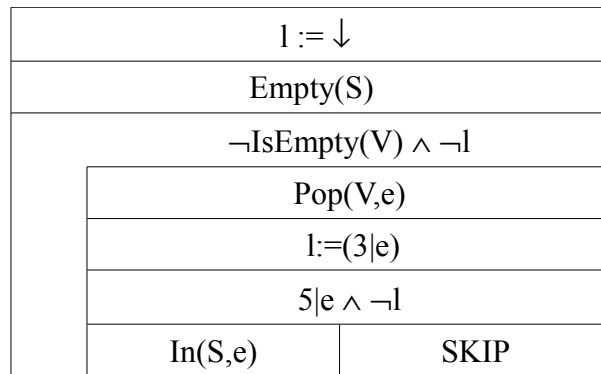
2. a) Jellemezzük az alábbi algoritmus műveletigényét!

A V verem kezdetben $n > 0$ számú elemet tartalmaz. Tekintsük meghatározó műveletnek a sor és verem adatszerkezet műveleteit és az oszthatóság vizsgálatát.

Hány meghatározó műveletet végez az algoritmus legjobb és legrosszabb esetben ($mM(n), MM(n)$)?

Mo: Legjobb esetben: $mM(n) = 1 + 2 + 1 + 1 + 1 = 6$ (Empty(S) (1x), ciklusfeltételben $\neg IsEmpty(V)$ (2x), 1-szer (Pop(V,e), 3|e, 5|e))

Legrosszabb esetben: $MM(n) = 1 + (n+1) + n*(1 + 1 + 1 + 1) = 5n+2$ (Empty(S) (1x), ciklusfeltételben $\neg IsEmpty(V)$ ((n+1)x), n-szer (Pop(V,e), 3|e, 5|e, In(S,e)))



b) Írd fel In(S,e) algoritmusát, ha az S sort láncolt adatszerkezettel reprezentáltuk!

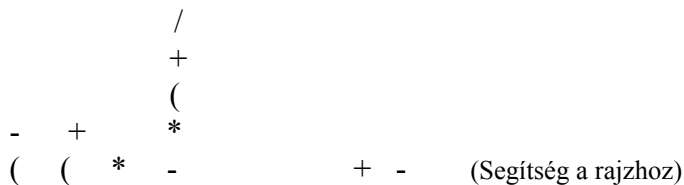
3. A vermes algoritmust alkalmazva határozd meg az alábbi kifejezés lengyelformáját!

Közben rajzold le a verem állapotait!

$$(a - b + c) * d - e * (f + 2 / g) + h - 2$$

Mo: a b - c + d * e f 2 g / + * - h + 2 -

Verem:



4. **a)** Add meg a főátlójában csak nullelemet tartalmazó, szimmetrikus mátrix sorfolytonos indexfüggvényét gazdaságos tárolás mellett! (A szimmetrikusan elhelyezkedő elemeknek csak az egyikét kell eltárolni. A nullelemet csak egy példányban kell eltárolni.)

$$\underline{M_0}: \text{ind } A[i, j] = \left\{ \sum_{k=1}^{i-1} k + j = \frac{(i-1) * i}{2} + j, \text{ ha } i > j \right.$$

$$\left. \frac{(j-i) * i}{2} + i, \text{ ha } i < j \right.$$

$$\left. \sum_{k=1}^{n-1} k + 1 = \frac{(n-1) * n}{2} + 1, \text{ ha } i = j \right.$$

$$\text{pl. } \begin{bmatrix} 0 & 1 & 2 & 4 \\ 1 & 0 & 3 & 5 \\ 2 & 3 & 0 & 6 \\ 4 & 5 & 6 & 0 \end{bmatrix}$$

b) Írj algoritmust, amely egy - a gyakorlaton vett módszerrel – láncoltan ábrázolt hiányos mátrixról eldönti, hogy szimmetrikus-e?

5. A Q sor egy $RR^{-1}S$ alakú karaktersorozatot tartalmaz (R^{-1} : az R szó visszafelé olvasva). Készíts olyan algoritmust, amely a Q sor tartalmát RRS-re változtatja! R és S hosszai ismertek. Az algoritmushoz (a Q soron kívül) csak 1 verem vagy 1 sor adatszerkezetet alkalmazhatsz. pl. baabház → babaház

6. Írj olyan algoritmust, amely egy input sorozatról megállapítja, hogy $(RR^{-1})^k S$ alakú-e, ahol $0 \leq k \in \mathbb{Z}$ és S az RR^{-1} kezdőszelete. R hossza ismert, de a k szám nem. Az algoritmushoz (2 vermet) vagy (2 sort) vagy (1 vermet és 1 sort) használhatsz. Megfelelő szó pl. **keddekkeddekkeddekkedd**

2. b) $In(s, e)$

```

new(p)
p^.elt := e
p^.next := NIL
IsEmpty(s)
S := p | last^.next := p
last := p | last := p
  
```

4. b)

```

l := igwer ; i := 1
i <= n and l
  r := S[i]
  c := O[i]
  r <= NIL and c <= NIL and l
    r^.oc > c
    l := (r^.oc = c^.sc and
    r^.sc = c^.oc and r^.elt =
    = c^.elt)
    r := r^.next
    c := c^.next
  l := (l and r = NIL and c = NIL)
  i := i + 1
Ret(l)
  
```

5.)

```

for i = 1 to |R|
  Out(Q, e)
  In(Q, e)
IsEmpty(V)
for j = 1 to |R|
  Out(Q, e)
  Push(V, e)
  not IsEmpty(V)
  Pop(V, e)
  In(Q, e)
for k = 1 to |S|
  Out(Q, e)
  In(Q, e)
  
```

(pl. 1 veremmel)

$check(S, V, e, l)$

```

not IsEmpty(V) and e <= l
  Pop(V, f)
  l := (e = f)
  Push(V, f)
  e := Read(S)
  
```

6.) (pl. 2 veremmel)

```

IsEmpty(V1) and IsEmpty(V2)
e := Read(S) ; i := 1
i <= |R| and e <= E
  Push(V1, e)
  e := Read(S) ; i := i + 1
  e <= E
    not IsEmpty(V1) and e <= E and l
      Pop(V1, f)
      l := (e = f)
      Push(V2, f)
      e := Read(S)
    not IsEmpty(V2) and e <= E and l
      Pop(V2, f)
      l := (e = f)
      Push(V1, f)
      e := Read(S)
  Ret(l)
  
```

checkat
+ i g g u e y e t :

→ check(~~S, V1, e, l~~)
S, V1, V2, e, l

→ check(~~S, V2, e, l~~)
S, V2, V1, e, l