

2.1a) func1(z) =

[9+7+8+6] = ~~15~~ ~~23~~ ~~31~~ ~~30~~

initialisiere ein Array S

für $i=1 \dots$ zeilen

für $j = \text{anfangsliste} \dots \text{endeliste}$

Index des z-Elements = int zaehler

Index des S-Elements = i

Wert des S-Elements = ~~z~~ Wert des z-Elements

if $S[\text{zaehler}] == 0$

füge das S-Element ein mit entsprechendem Pointer
esle

füge das S-Element am Ende ein mit Pointer

Idee: Man geht die Elemente der Zeilen darstellung
durch und tauscht den Index des Elements
mit Arrayindexen.

Am Ende werden die Elemente noch mit Pointern
verkettet

~~func1(z) =~~
~~...~~
~~...~~

Laufzeit?

- 1p.

b) für $i=1 \dots n$

für $j=1 \dots n$

$a = A[i]$ // geht die Elemente der i -Zeile durch

$s = 0$ // int, berechnet den Wert für $C[j]$

while $a = A.next$ // solange es ein Nächstes gibt

if $B[j].index == i$

$s += B[j].wert * a.wert$

$C[j]$ append $((j, s))$ // hänge ein Element in die Ergebnisliste

für $j=1 \dots n$

if $B[j].index == i$

$B[j].delete First()$ // entferne die erste Spalte von B

Laufzeit $O(n^3)$

2.4a)

(<) >

b) Lösung per Stack, (gelesen von links nach rechts)

für jedes "(" : push "("

für jedes "<" : push "<"

für jedes ")" : pop "("

für jede ">" : pop "<"

syntaktisch falsch - 1 P.

sollte der Stack am Ende des Wortes nicht-leer sein,

oder ein zu "pop"-endes Zeichen

nicht oben auf dem Speicher liegen,

gebe NICHT-LEGAL aus, LEGAL sonst.

Laufzeit $O(n)$

2.3) Datenstruktur = Queue

enqueue v // pack die root in die Queue

solange (queue $\neq 0$) // solange die Queue nicht leer ist

enqueue v.left, falls v.left $\neq 0$

enqueue v.right, falls v.right $\neq 0$

print (dequeue) // gib das unterste Element der Queue aus

2.4) Es sei C ein leeres Array von Knoten

// hier werden alle Knoten abgelegt, die von

u zur Wurzel durchlaufen werden

solange B[u] $\neq 0$

füge B[u] in C hinzu

B[u] = B[B[u]] // gehen zur Wurzel

solange B[v] $\neq 0$

if B[v] in C enthalten // Schnittpunkt gefunden

gebe B[v] aus; return

else

$B[v] = B[B[v]]$ // weiter zur Wurzel