

PALINDROME

① palindrome (L):
rev (L, L).

② palindrome (L):
reverse (L, R),
 $L = R$.

③ palindrome ($[T]$).
palindrome ($[X]$ ~~not~~).
palindrome ($[A, T]$):
conc ($T, [H], T_2$),
palindrome (T_2).

④ maketto conc ($T, [H], T_2$)

bo kdo postavlja \geq last (T, \bar{H})

Da je... (a kako ugotoviti, kje
ne gre! je sredina...)

zgut: "Z last/2 bo težko to implementirati,
postavši moje s conc/3"
vec → kar da je zadnji element?

ZANIMIVOST
PA VERTIJA je na koncu
predstavljan kot zanimivost!
(brez manjgov o nej prej)

- * če res je tako pa
morda dobijo naslednje ali
nereg... konci?
- * kaj pa "skoraj" super rezultat?
palindrome (L):
reverse (L, R)
 $L = R$.
→ To je treba zagotoviti

① NAMIG: "To se da ne konča,
za dočki je ena samo
s premenljivkami;"

② manjka [x] ⁽²⁾ lot noben pogoj,
a je [x, x], [x, x]
(morda spomnimo?)

③ DVA ROBNA POGOJKA

če en manjka (vse kriterij)

(a) dano protipriber!

(b) In, če te vedno isto manjka,
tečnik "kaj pa je sedaj" [a, a]
ali [a, b, a])
odvisno od prvega je
kateri manjka

④ Napacen noben pogoj
palindrome ($[X]$ ~~not~~)
palindrome ($-$).

Hint: " - ne potem poljubnega
seznam z enino elatom, D
ampak poljubno spremenljivo
ki je lahko karali, tudi pogrešno...
npr.

→ vec: seznam z elementom se zoperstreže le tako?
npr. (poljubni)

spletivi nasveti

① STAVAN
Rekurzivni korak je naključno tabula

☒

1	1	1	1	1
---	---	---	---	---

 (7)

pravim:
 $x = y$; ↓ rekurzija;
↓ prevent sredine ;)
yes/no

→ vec: Kako dobim zadnji element? ;)

② "Če je ~~not~~ ^{AH} prvi element ukratko
seznam in ~~not~~ ^{AH} zadnji element
seznam ter je
vsebuje (sredji ~~not~~ del seznam
(prvega) palindrom,
potem je tudi cel seznam palindrom
sicer pa ni."

④ Palindrom je seznam, ki
se bere od spredaj in
z dolje nazaj.

(spomni manjši rez je palindrom)
npr. [a, b, c] ali [c, b, a]
TAJACRNO NB:

Palindrom je seznam, ki se
obravičen bere nako két narednega.

SHIFT LEFT

$\text{shiftleft}([H|T], L2) := \text{conc}(T, [H], L2).$

(*) $\text{conc}(T, H, L2).$
* generični

(2) neporavnava $L2$ it cone
in $L2$, ki si vraci v glavi...
(tole je idealno za AT)

(3) Tukaj nobenega rekurzije pogojja pravzaprav ni.
(zato kaže se strive $\text{conc}/3$)

A poskrbimo negotovosti, da v
stavki ni ve rekurzije in
ne $\text{conc}/3$, poleg je to najbolj
"prsti" merekuven stavek...
Kaj pa bi bil manjši?
(norda kar lahko it splošnih manjših?)

(4) ~~$L2 = [T | H]$~~

(a) napisi reverz
konic ni drža (slab) macina;
poslus mordeti Heutejo, ke delo
gačen do konca in potem delo
na koncu
vsego enkratnevaje, sestava
gačen do konca in potem delo
na koncu

-: (2) $\text{shiftleft}(L2, L1)$

-: (4) $\text{shiftleft}(L1, [H | L2])$

$\text{shiftleft}(L1, L2) :=$

$L1 = [H | T],$

...

* generični, lepotni

\Rightarrow kaj pa tale (pravila) nista?

$\text{shiftleft}([X], [X]).$

$\text{shiftleft}([H1, H2 | T], [H2 | R]) := \rightarrow$ vec: Kako dodajam na konec članaka?

$\text{shiftleft}([H1 | T], R).$

"prvi pri element po seznamu
na konec ..."

(*) mordetljevje ...

Kako to postvari?

enkratno operacija.

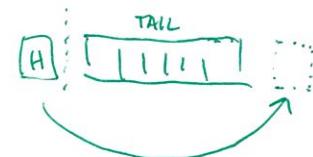
shiftleft, operacija za

če je vrednostna, da ne

čini drugi mordetljiv

(1)

Preprosto! ... v seznam vraken
pri element in ga dodam
(v preostalih) na konec.
rečimo T



(2)

"če seznam je seznam L
restavrijen it glave H in nepri T
in de H dodams na konec
T, potem je rezultat seznam
"zadnjem" or levo".

(*) ali je pri shiftleft #4)

$L = [T | H]$

pri shiftleft

#2 ... glej

Tukaj, ko imamo

idemo za A.I.

ter L1 in L2, (v (1))

izpolnjujemo L2 in L2

(ne dobi T)

je v tem rezultatu stan

pozne zadaj! izvor!

"Kaj mi de, ker samo

operacija shiftleft

rezultat #2 (→)

shiftleft, korej:

shiftleft je operacija za

seznam operacije za

seznam operacije za

seznam operacije za

je [E | T] ali "shiftleft" seznam L

če je T seznam brez + taga

L in je E takoj! celotni seznam

če je E takoj! celotni seznam



(2)

U seznam pollice

in seznam stran zadevi elemt

ter ga preobstavlju da

če je seznam pollice

če je seznam stran zadevi elemt

splošni nasveti

SHIFTLGH

SUBLIST

* (to je v listu "tricky" uporabite conc/3)

sublist(L, SL):-
conc(-, L1, L),
conc(SL, -, L1).

(Ta rečišča večkrat ponovi
pravzaprav letnem Č) kot rečišča,
a mas to ne more preneči.)

①

v listu ni (prave) rekurrense

HINT: "Sublist lahko definirati
posredno rekurzivno"
~ uporabi conc/3

* glavnico: delekuj, da
je uporabila uporabile rekurrense!

* ni problem
začetke,

good!

(uporablja mat conc/3!
az. append/3!)

② Ko enkrat Student pogrunci sam

ali vidi nasvet o reševanju s conc/3,
se problem prevede na pravilno
"vetačo" spremenljivko...
... tu je idejna različica za AI. način od Ti.
(najbrž preveč kombinacij (napačnih), da
bi jih bilo smiselnovoriti en po enu...)

* mora biti zadovoljstvo:

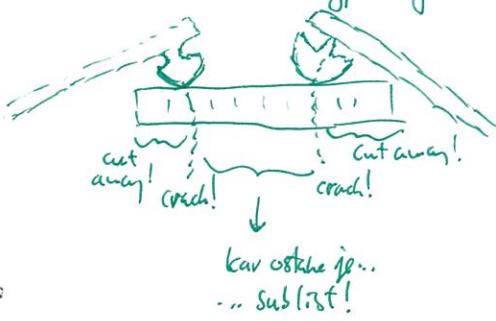
ale je do tu že
pravna ideja
kakstradiga
az. vraciš pa
različne reči?/
(redko)

③

Najprej razloži kaj sploh
naloge zahteva... ve podrazumevajo,
am pa kar podsetim (če elementi
se držijo skupaj).

④

visualni limit: "sečaj, sečaj"!!



kar ostanejo...
... sublist!

→ več: Kako "odslečati" del listnega?

sprejti: conc(-, Ostanek, Sečenje)

zadaj: conc(Ostanek, -, Sečenje)

④

"- Če danes utranno L odsekam
naj naj na začetku in potem ...
naj naj na koncu, mi ostane
navadno podsekhan..."