

3. naloga: Dan je naslednji problem:

Vhod: Zaporedje števil $A = [a_1, a_2, \dots, a_n]$ in vrednost v . *Izhod:* Indeks i , če je $v = A[i]$, ali posebni znak NILL, če v ni vsebovan v A .

- (i) Napišite algoritem za linearno iskanje, ki se sprehodi čez zaporedje in išče v .
- (ii) Z uporabo zanke invariante pokažite pravilnost algoritma.

Algoritem za linearno iskanje:

```
def LinearSearch(A, v):
    i = 0
    n = len(A)
    while i < n-1:
        if v == A[i]:
            return i
        i++
    return NILL
```

Algoritem deluje tako, da se sprehodi čez seznam A in vsako iteracijo zanke preveri, ali je trenutni element $A[i]$ enak podani vrednosti v . Ustavi se, ko najde vrednost $A[i]$, ki je enaka v , ali pa, ko $i \geq n - 1$.

Zančna invarianta za *LinearSearch*:

“Zančna invarianta je lastnost algoritma, ki velja pred začetkom izvajanja, med izvajanjem ter po koncu izvajanja algoritma.”

Zančna invarianta: Na začetku vsake iteracije podseznam $A[0 : i - 1]$ ne vsebuje vrednosti v .

1. Pred začetkom izvajanja zančna invarianta velja, saj podseznam $A[0 : i - 1]$ ne vsebuje vrednosti v , saj je tak podseznam prazen.
2. Vsako iteracijo preverja, ali je $v = A[i]$; če je, se izvajanje ustavi in vrne se indeks i - v drugem primeru invarianta drži, saj tudi v tem primeru podseznam $A[0 : i]$ ne vsebuje elementa v . Po primerjavi, se i spet poveča in nastopi nova iteracija, ko spet velja, da podseznam $A[0 : i - 1]$ ne vsebuje v .
3. Zanka se ustavi, ko $i = n = \text{len}(A)$, kar zagotovi, da v podseznamu $A[0 : n - 1]$ (celotni seznam) ne vsebuje vrednosti v . Če se algoritem ustavi preden najde v , vemo, da elementa ni v seznamu - vrne NILL.