

Drevo predpon

Luka Toplak in Anže Ozimek

Fakulteta za matematiko in fiziko
Univerza v Ljubljani

15. december 2021

Motivacija za uporabo podatkovne strukture

- Analiziramo besede v danem besedilu

Motivacija za uporabo podatkovne strukture

- Analiziramo besede v danem besedilu
- Leksikografska urejenost

Motivacija za uporabo podatkovne strukture

- Analiziramo besede v danem besedilu
- Leksikografska urejenost
- Besede poskušamo shranjevati čim bolj kompaktno

Motivacija za uporabo podatkovne strukture

- Analiziramo besede v danem besedilu
- Leksikografska urejenost
- Besede poskušamo shranjevati čim bolj kompaktno
- Dostop do iskane besede mora biti hiter

Rešitev problema s pomočjo seznama

- Besede iz besedila vstavimo v seznam

Rešitev problema s pomočjo seznama

- Besede iz besedila vstavimo v seznam
- Uredimo seznam

Rešitev problema s pomočjo seznama

- Besede iz besedila vstavimo v seznam
- Uredimo seznam
- Odstranimo ponovljene besede

Rešitev problema s pomočjo seznama

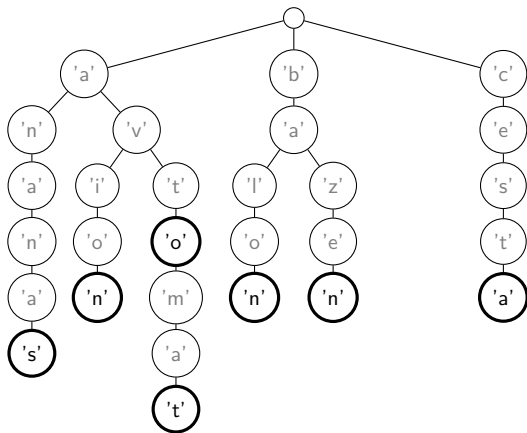
- Besede iz besedila vstavimo v seznam
- Uredimo seznam
- Odstranimo ponovljene besede
- Besede iščemo z bisekcijo

Rešitev problema s pomočjo seznama

- Besede iz besedila vstavimo v seznam
- Uredimo seznam
- Odstranimo ponovljene besede
- Besede iščemo z bisekcijo

[... , "avto", "avion", "avtomat", ...]

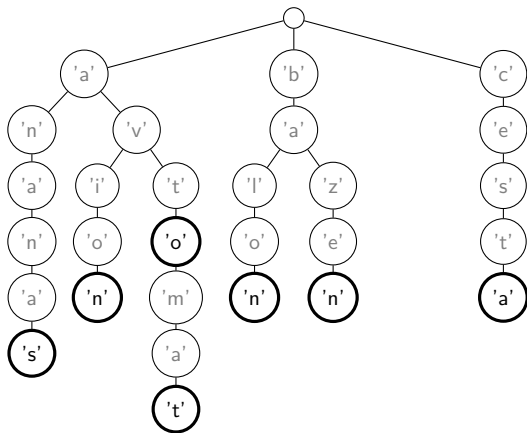
Drevo predpon



"bazen", "ananas", "cesta", "balon",
"avto", "avion", "avtomat"

Drevo predpon

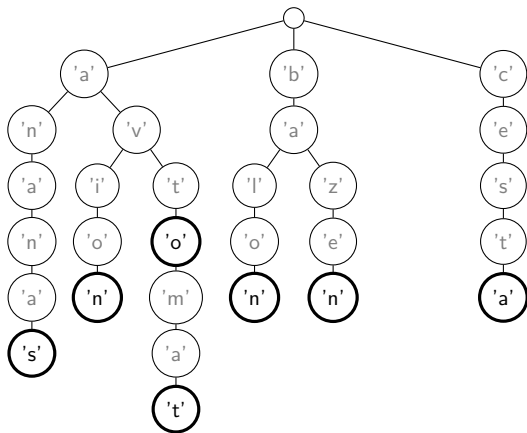
- Iskalno drevo, ki v vozliščih hrani elemente nizov



"bazen", "ananas", "cesta", "balon",
"avto", "avion", "avtomat"

Drevo predpon

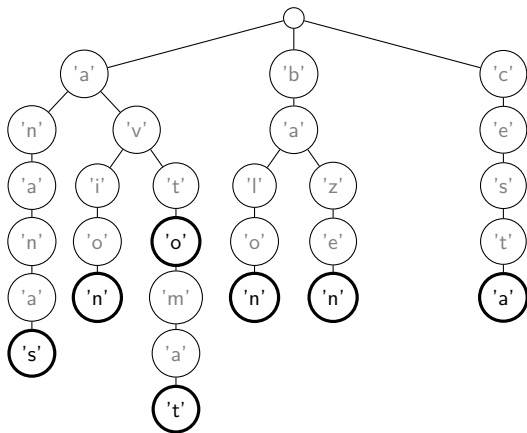
- Iskalno drevo, ki v vozliščih hrani elemente nizov
- Pot od korena do vozlišča predstavlja niz



"bazen", "ananas", "cesta", "balon",
"avto", "avion", "avtomat"

Drevo predpon

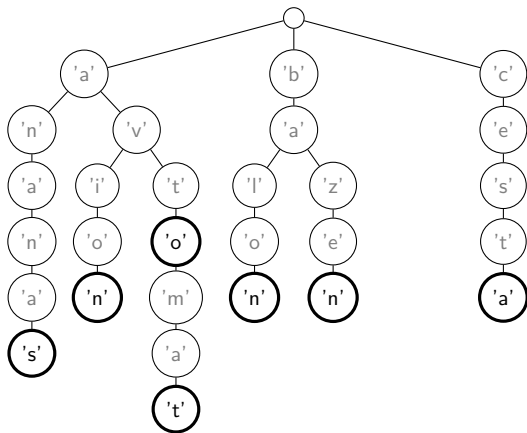
- Iskalno drevo, ki v vozliščih hrani elemente nizov
- Pot od korena do vozlišča predstavlja niz
- Vozlišče, ki predstavlja zadnji element niza je označeno



"bazen", "ananas", "cesta", "balon",
"avto", "avion", "avtomat"

Drevo predpon

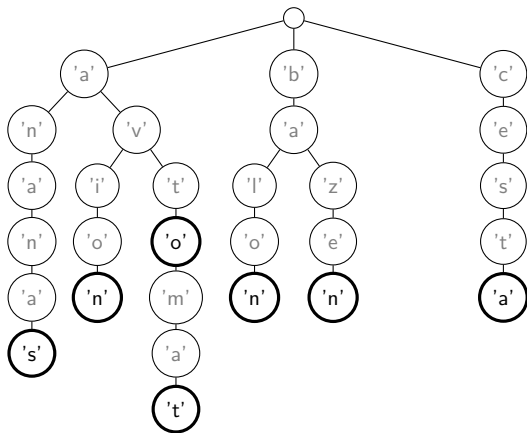
- Iskalno drevo, ki v vozliščih hrani elemente nizov
- Pot od korena do vozlišča predstavlja niz
- Vozlišče, ki predstavlja zadnji element niza je označeno
- Potomci vsakega vozlišča so urejeni



"bazen", "ananas", "cesta", "balon",
"avto", "avion", "avtomat"

Drevo predpon

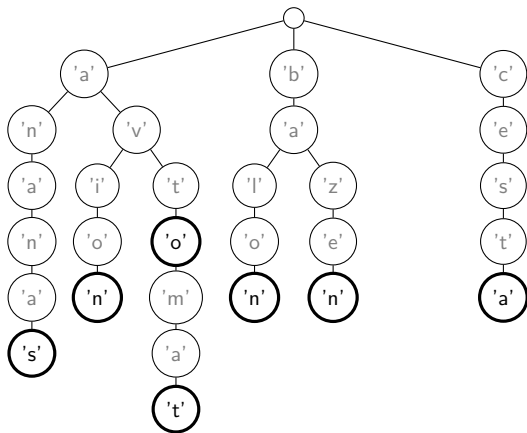
- Iskalno drevo, ki v vozliščih hrani elemente nizov
- Pot od korena do vozlišča predstavlja niz
- Vozlišče, ki predstavlja zadnji element niza je označeno
- Potomci vsakega vozlišča so urejeni
- Koren predstavlja prazen niz



"bazen", "ananas", "cesta", "balon",
"avto", "avion", "avtomat"

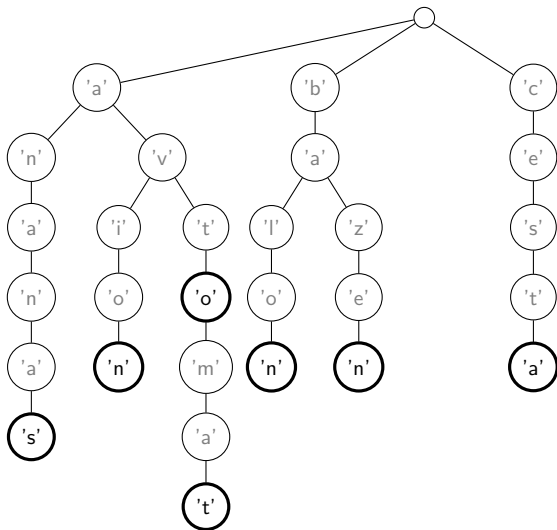
Drevo predpon

- Iskalno drevo, ki v vozliščih hrani elemente nizov
- Pot od korena do vozlišča predstavlja niz
- Vozlišče, ki predstavlja zadnji element niza je označeno
- Potomci vsakega vozlišča so urejeni
- Koren predstavlja prazen niz
- Omejimo se na nize znakov

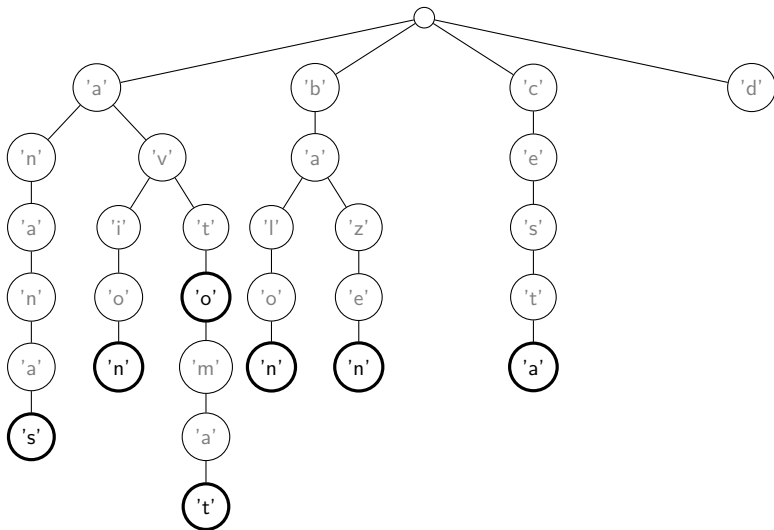


"bazen", "ananas", "cesta", "balon",
"avto", "avion", "avtomat"

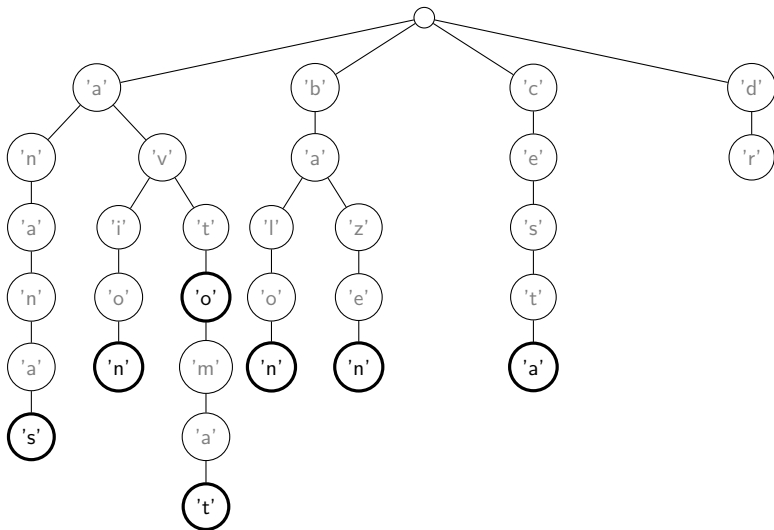
Vstavljanje besede "drevo"



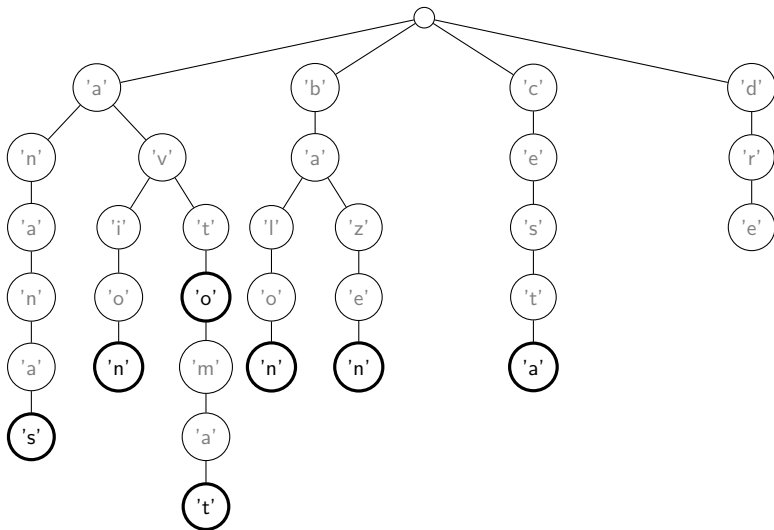
Vstavljanje besede "drevo"



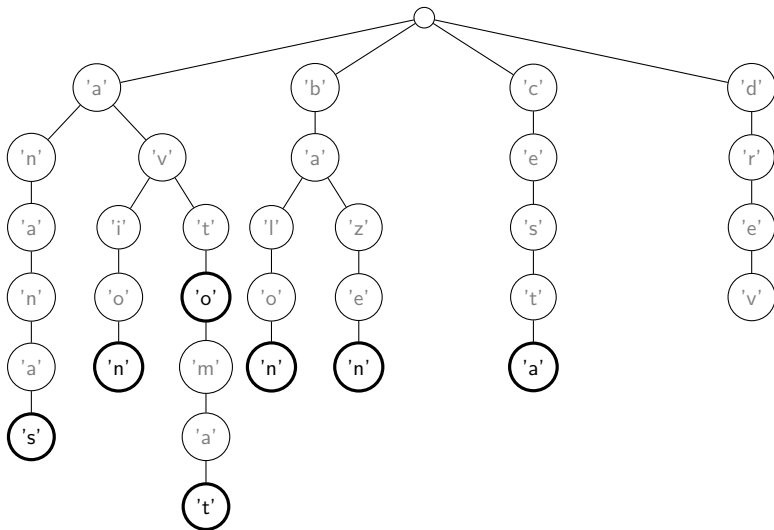
Vstavljanje besede "drevo"



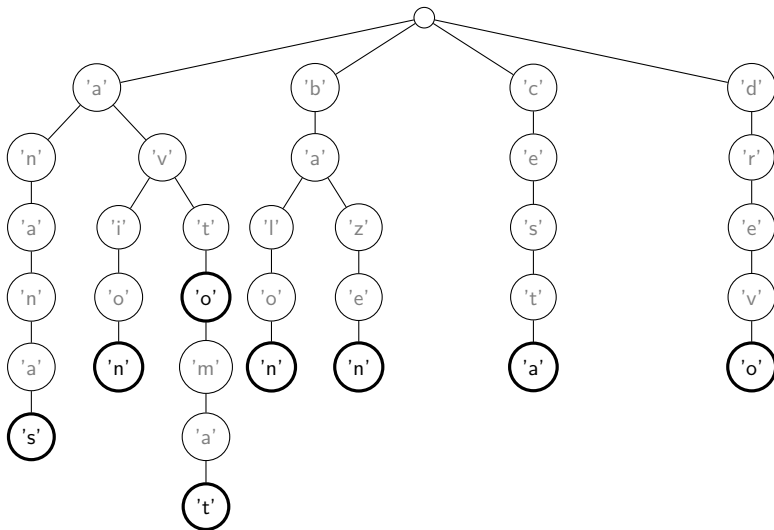
Vstavljanje besede "drevo"



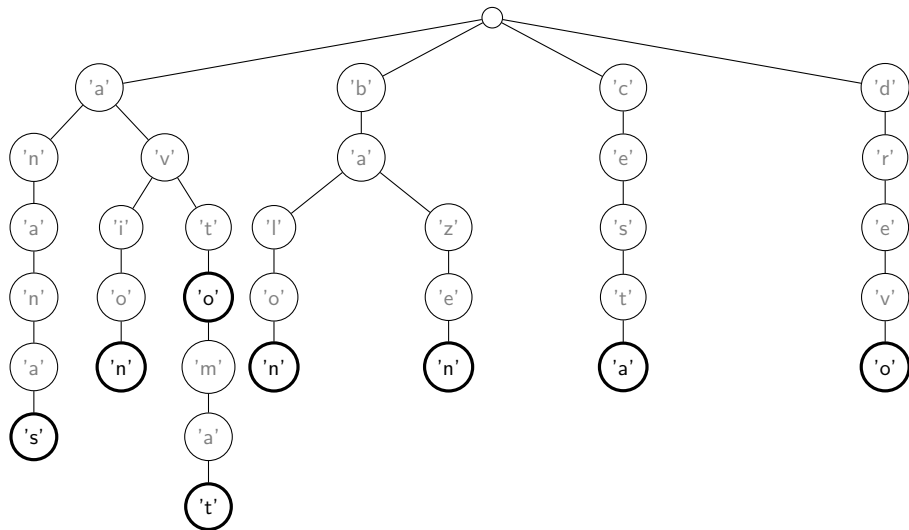
Vstavljanje besede "drevo"



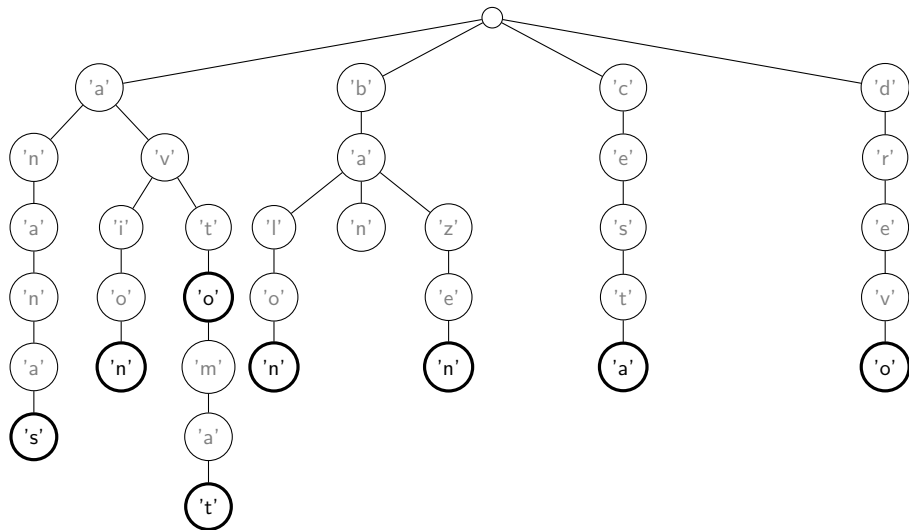
Vstavljanje besede "drevo"



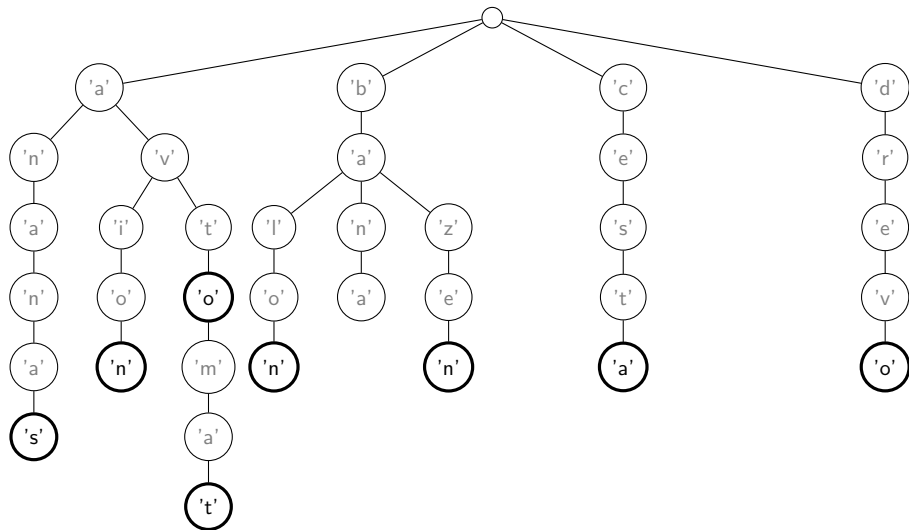
Vstavljanje besede "banana"



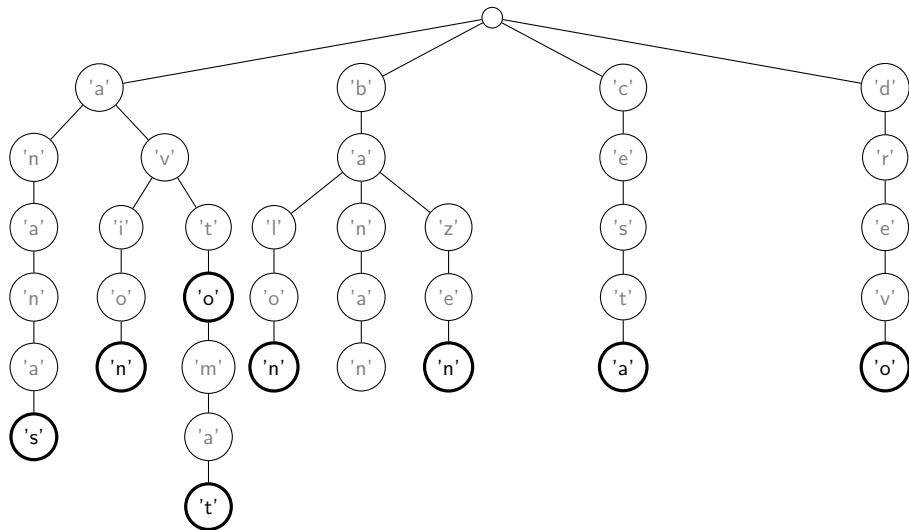
Vstavljanje besede "banana"



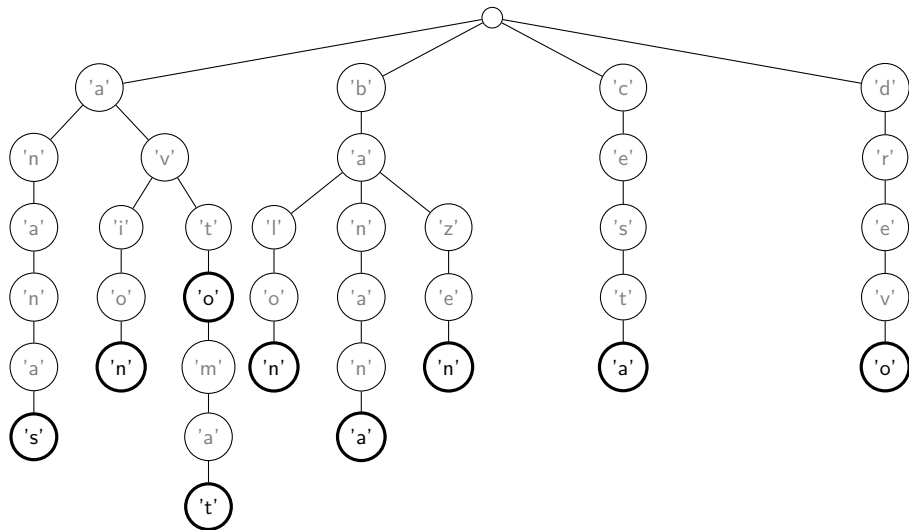
Vstavljanje besede "banana"



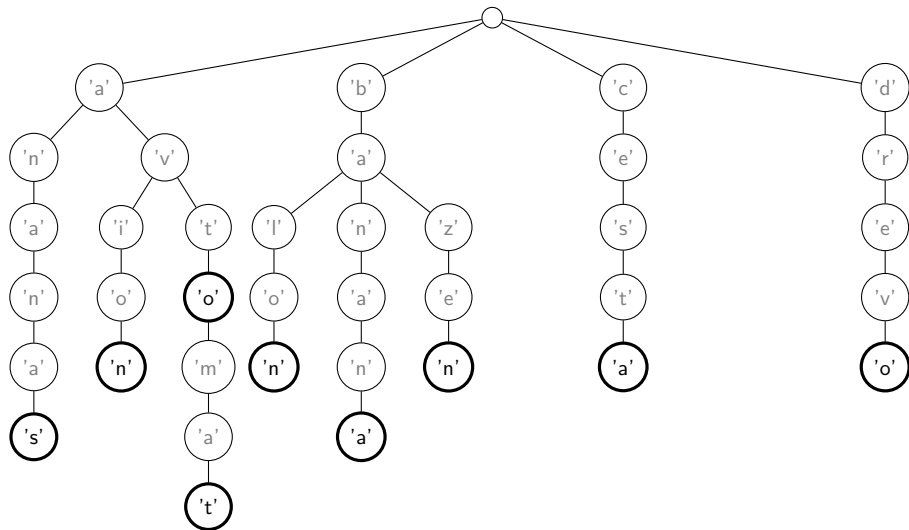
Vstavljanje besede "banana"



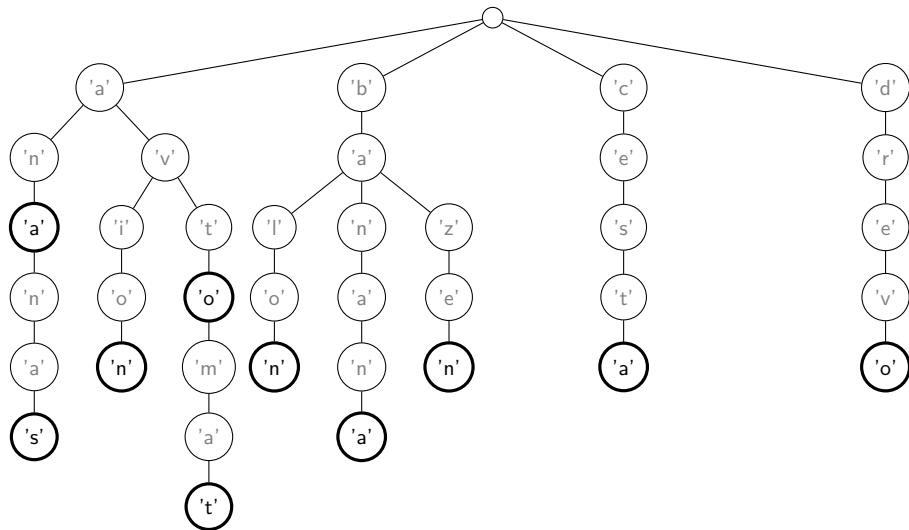
Vstavljanje besede "banana"



Vstavljanje besede "ana"



Vstavljanje besede "ana"



Vstavljanje

```
procedure VSTAVI(drevo_predpon, beseda)
    trenutno  $\leftarrow$  drevo_predpon.koren
    for each z  $\leftarrow$  znak v besedi beseda do
        najden  $\leftarrow$  false
        mesto  $\leftarrow$  0
        for each p  $\leftarrow$  potomec vozlišča trenutno do
            if z = p.vrednost then
                najden  $\leftarrow$  true
                break
            else if z < p.vrednost then
                break
            mesto  $\leftarrow$  mesto + 1

    if not najden then
        novo_voz  $\leftarrow$  VOZLIŠČE(z)
        trenutno.potomci.VSTAVI__NA(mesto, novo_voz)
    trenutno  $\leftarrow$  trenutno.potomci[mesto]
    trenutno.je_beseda  $\leftarrow$  true
```

- Po drevesu se premikamo skupaj z znaki v nizu

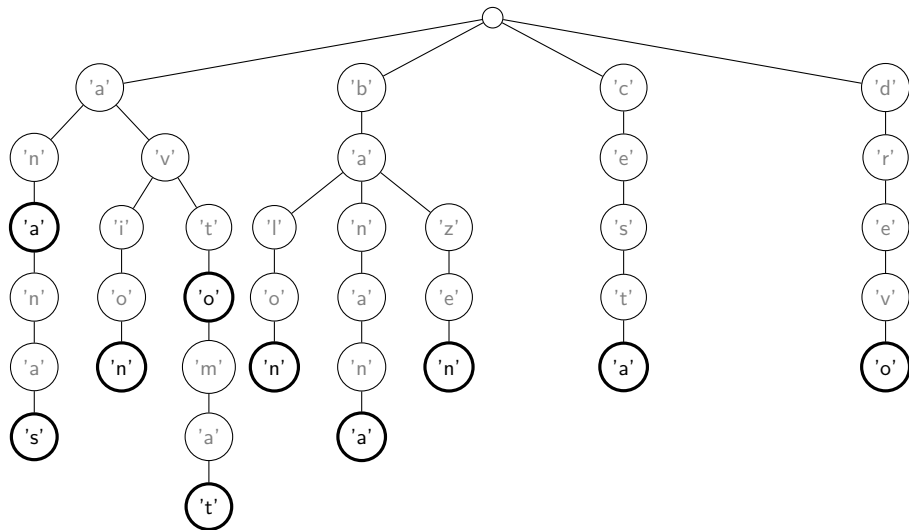
- Po drevesu se premikamo skupaj z znaki v nizu
- Če med potomci ne najdemo trenutnega znaka, besede ni v drevesu

- Po drevesu se premikamo skupaj z znaki v nizu
- Če med potomci ne najdemo trenutnega znaka, besede ni v drevesu
- Če pridemo do konca besede, preverimo ali je trenutno vozlišče označeno

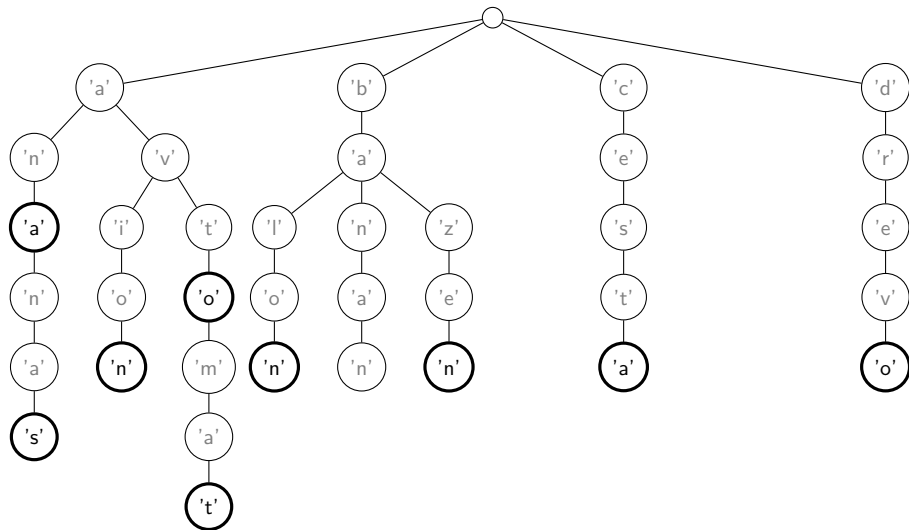
- Potrdimo, da je beseda v drevesu

- Potrdimo, da je beseda v drevesu
- Na poti nazaj brišemo vozlišča, dokler ne naletimo na drugo besedo ali vejitev

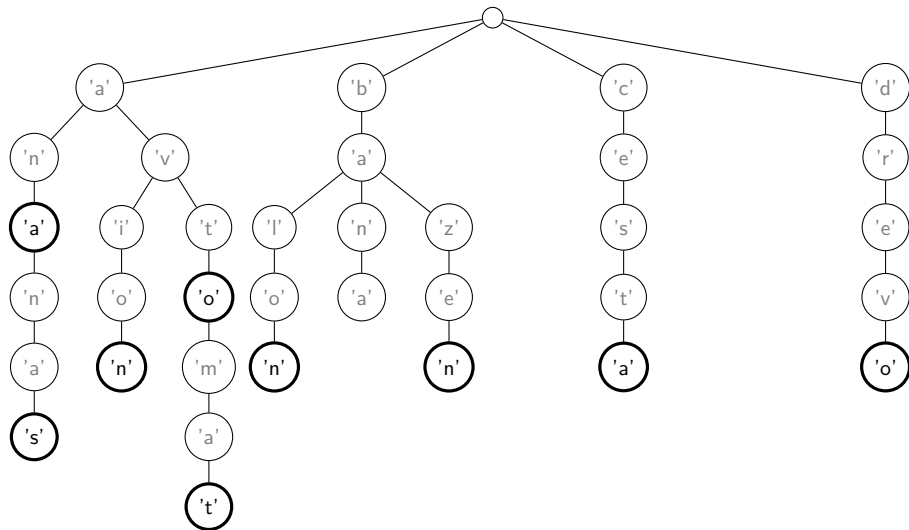
Brisanje besede "banana"



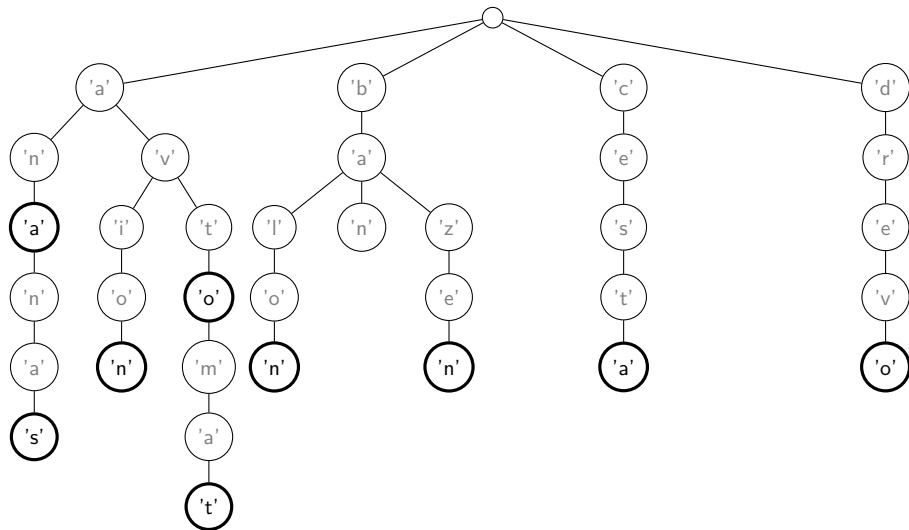
Brisanje besede "banana"



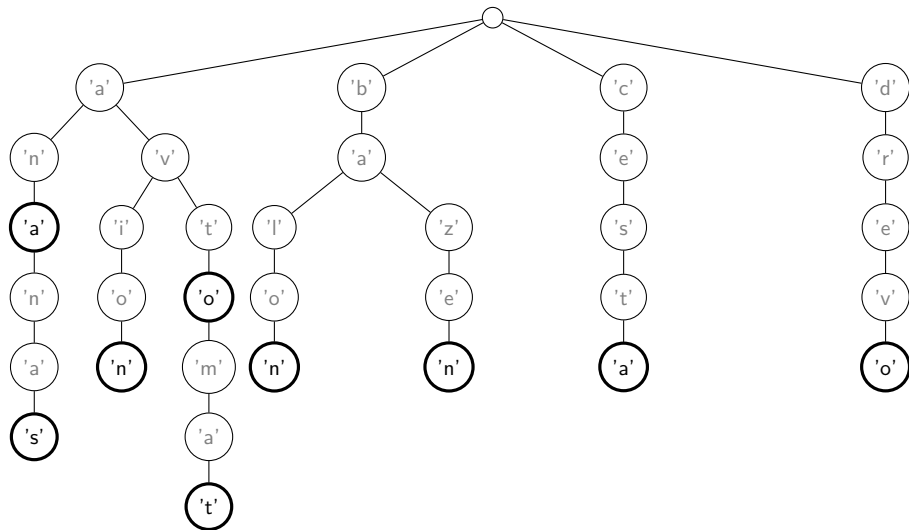
Brisanje besede "banana"



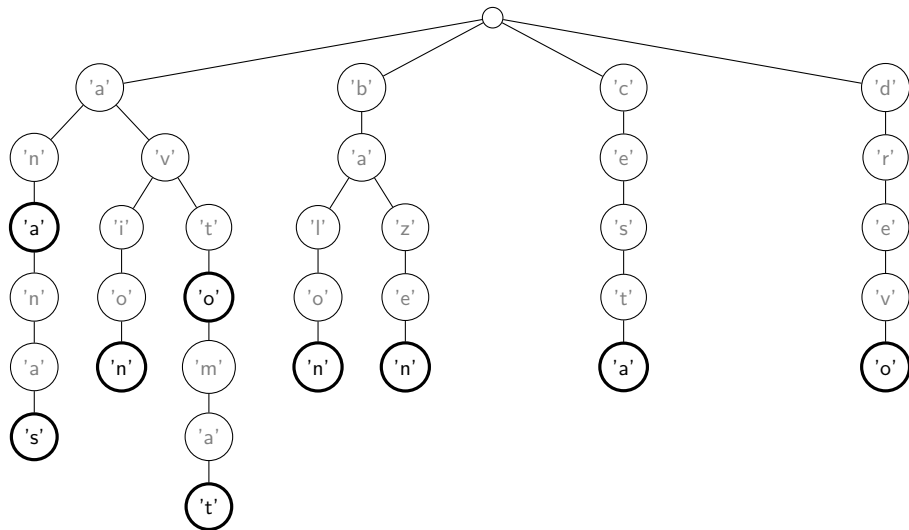
Brisanje besede "banana"



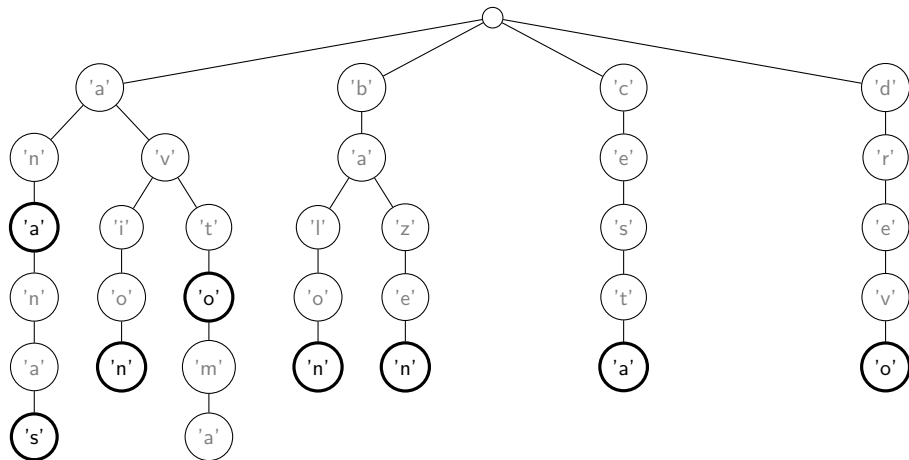
Brisanje besede "banana"



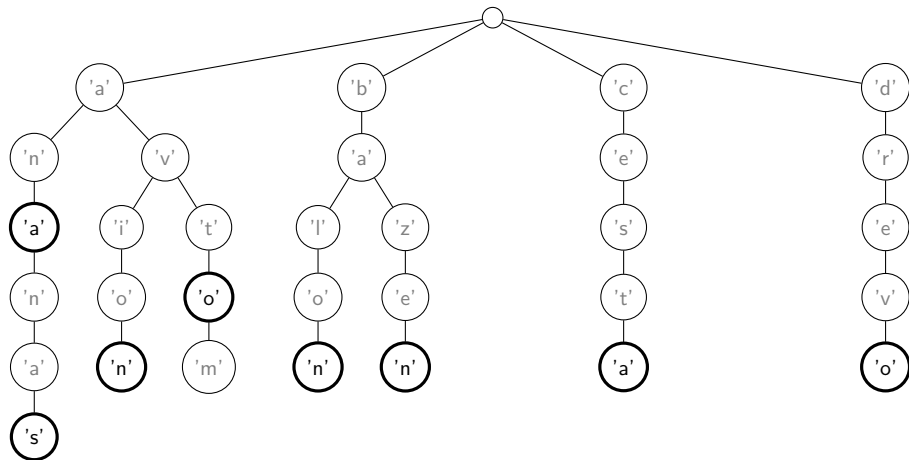
Brisanje besede "avtomat"



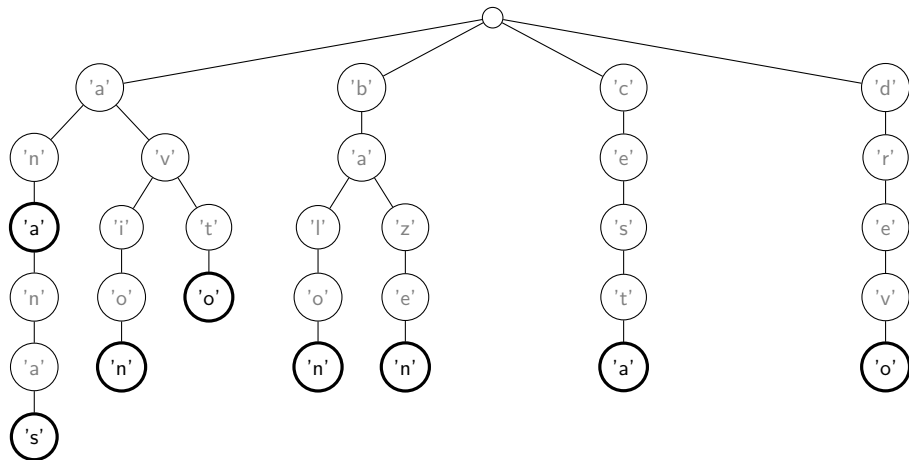
Brisanje besede "avtomat"



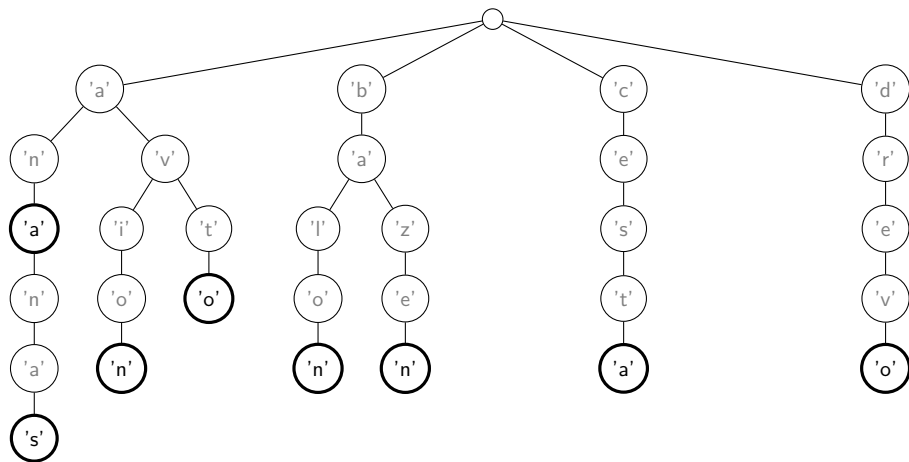
Brisanje besede "avtomat"



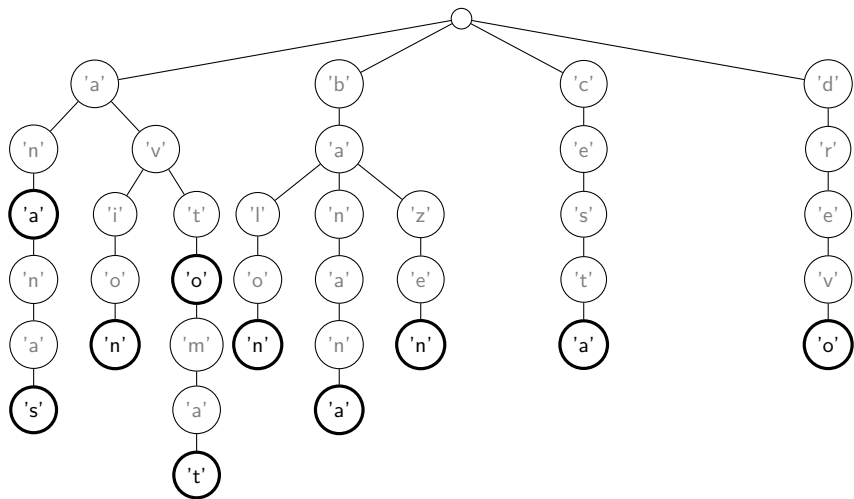
Brisanje besede "avtomat"



Brisanje besede "baza"



Primer urejanja



["ana", "ananas", "avion", "avto", "avtomat",
"balon", "banana", "bazen", "cesta", "drevo"]

- Premi pregled

Urejanje

- Premi pregled
- Med sprehtodom posodabljamu besedo

Urejanje

- Premi pregled
- Med sprehodom posodabljam besedo
- Shranimo besede na vozliščih, ki so označena

Urejanje

- Premi pregled
- Med sprehomod posodabljam besedo
- Shranimo besede na vozliščih, ki so označena

```
procedure UREJENE(trenutno_voz, trenutna_beseda)
    besede ← prazen seznam nizov
    if trenutno_voz.je_beseda then
        besede.DODAJ_NA_KONEC(trenutna_beseda)

    for each p ← potomec vozlišča trenutno_voz do
        ost_besede ← UREJENO(p, trenutna_beseda + p.vrednost)
        besede.DOPOLNI(ost_besede)           ▷ Združimo seznama

    return besede
```

Uporaba – “autocomplete”

- V drevesu predpon so že uporabljene besede

Uporaba – “autocomplete”

- V drevesu predpon so že uporabljene besede
- Med vnosom besede se premikamo po drevesu

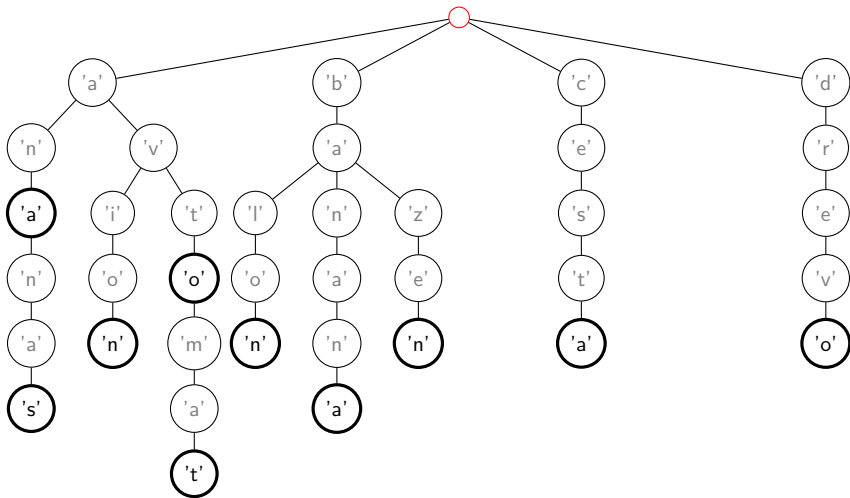
Uporaba – “autocomplete”

- V drevesu predpon so že uporabljene besede
- Med vnosom besede se premikamo po drevesu
- Na vsakem koraku prikažemo besede z napisano predpono

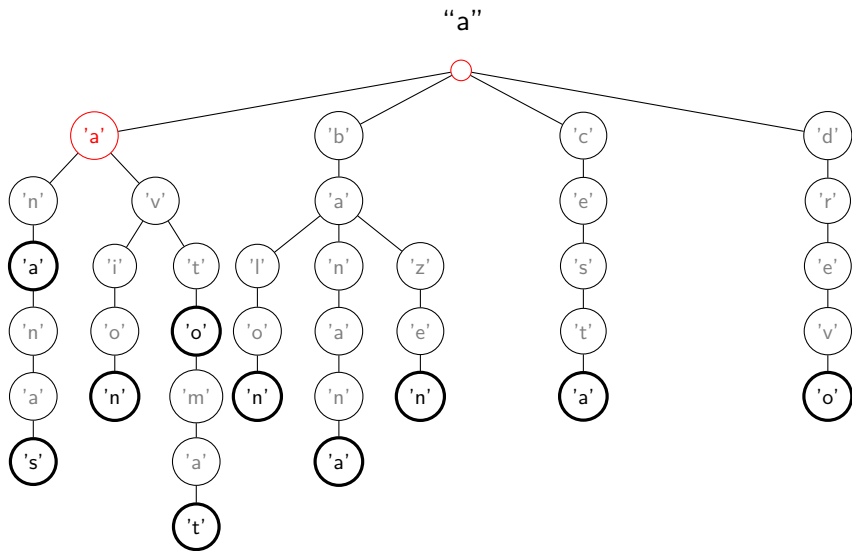
Uporaba – “autocomplete”

- V drevesu predpon so že uporabljene besede
- Med vnosom besede se premikamo po drevesu
- Na vsakem koraku prikažemo besede z napisano predpono
- Če uporabnik vnese besedo, ki je še ni, jo dodamo

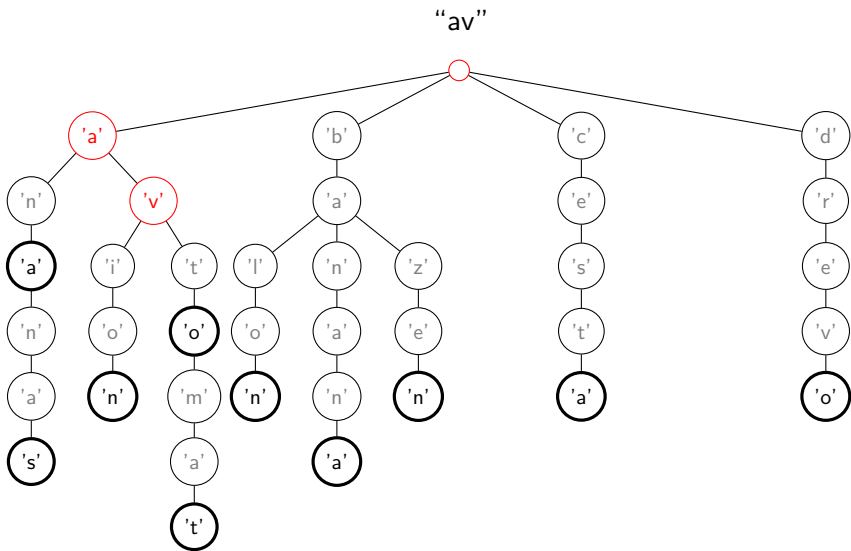
“ ”



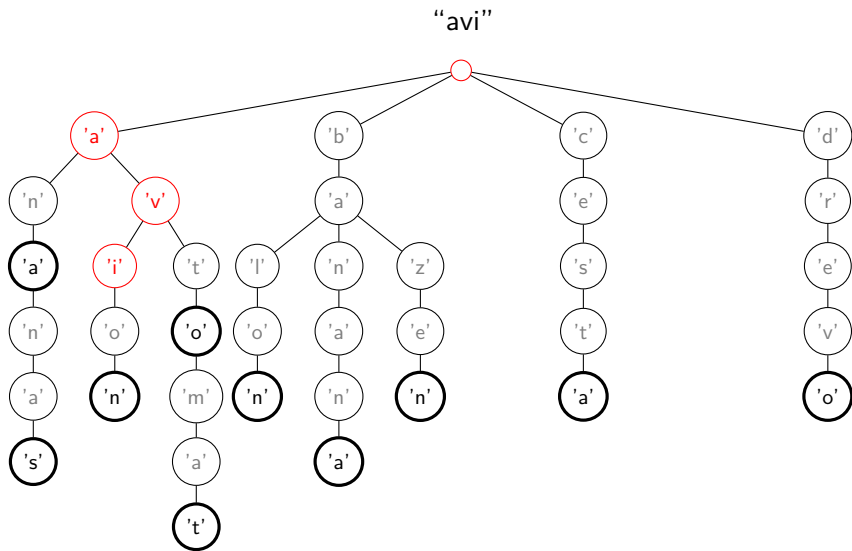
["ana", "ananas", "avion", "avto", "avtomat",
"balon", "banana", "bazen", "cesta", "drevo"]



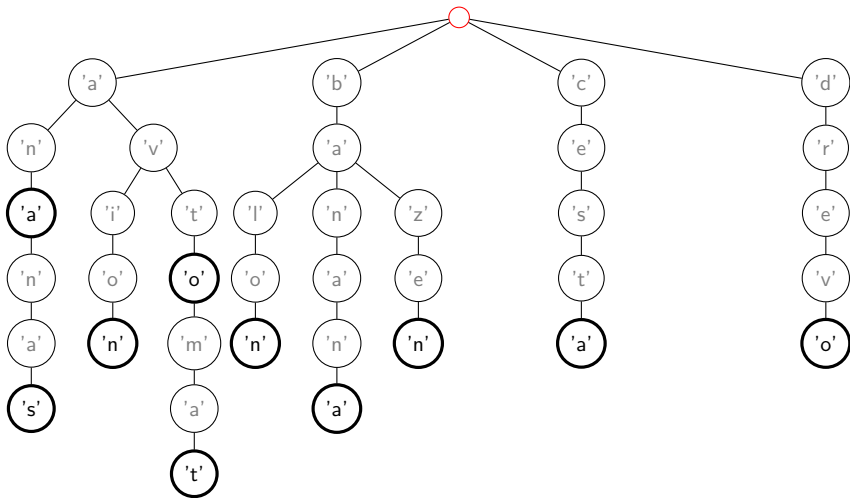
["ana", "ananas", "avion", "avto", "avtomat"]



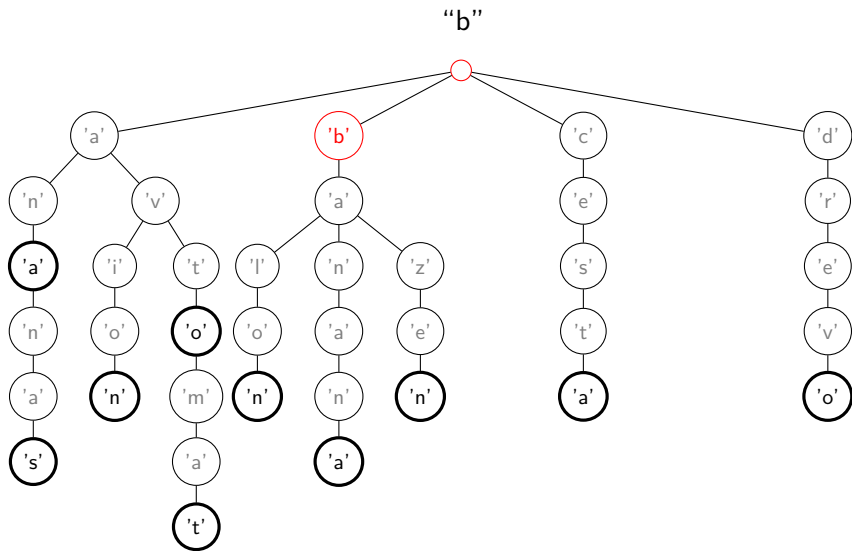
["avion", "avto", "avtomat"]



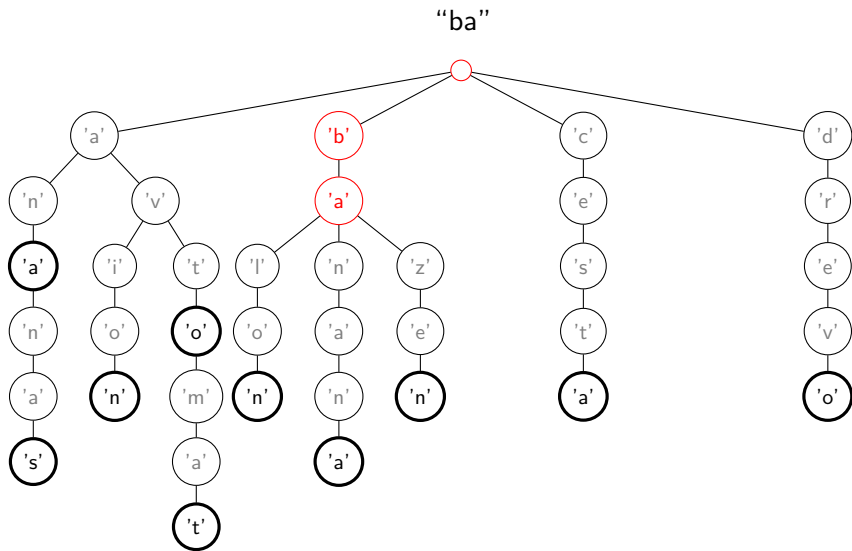
“ ”



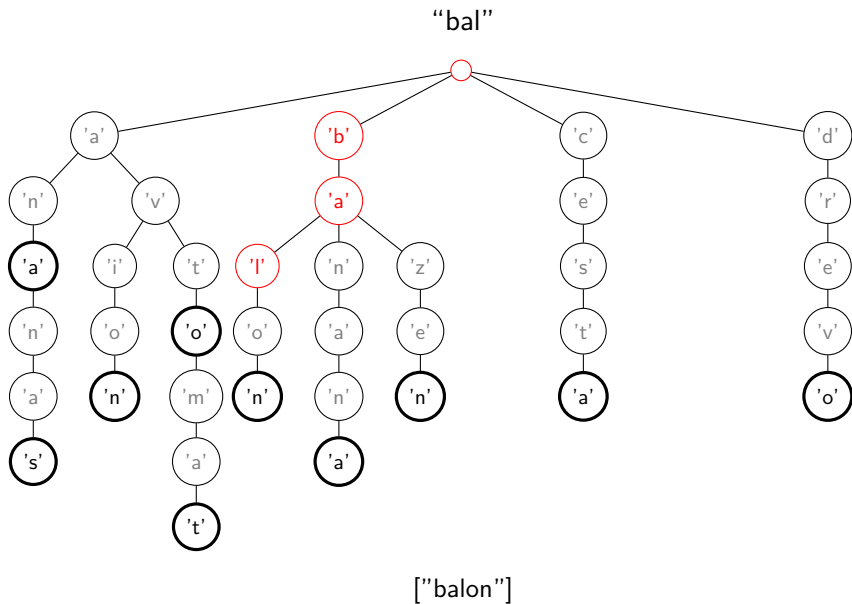
["ana", "ananas", "avion", "avto", "avtomat",
"balon", "banana", "bazen", "cesta", "drevo"]



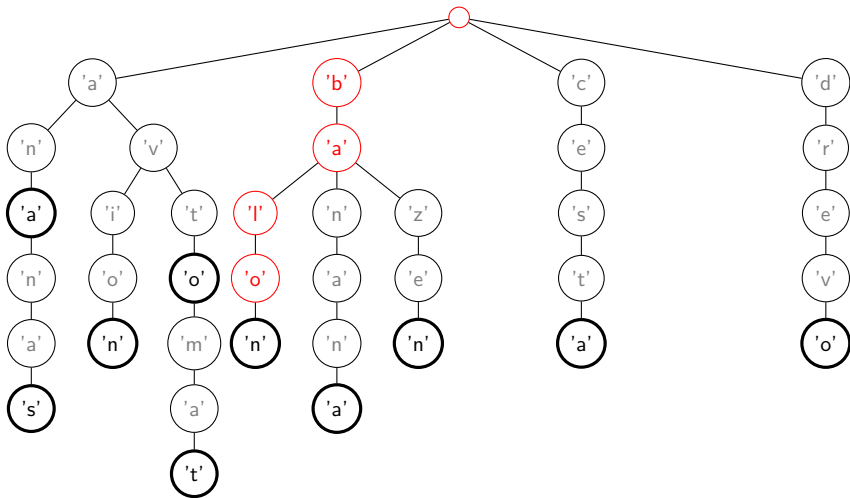
["balon", "banana", "bazen"]



["balon", "banana", "bazen"]



"balo"



["balon"]

- Osnovna operacija je premik na nižji nivo

Časovna zahtevnost

- Osnovna operacija je premik na nižji nivo
- Osnovne operacije na drevesu
 - Velikost problema je dolžina niza – k
 - Vstavljanje, brisanje, iskanje – $\mathcal{O}(k)$

Časovna zahtevnost

- Osnovna operacija je premik na nižji nivo
- Osnovne operacije na drevesu
 - Velikost problema je dolžina niza – k
 - Vstavljanje, brisanje, iskanje – $\mathcal{O}(k)$
- Urejanje
 - Velikost problema sta število besed in dolžina najdaljše besede – n in l
 - $\mathcal{O}(n \cdot l)$

Časovna zahtevnost

- Osnovna operacija je premik na nižji nivo
- Osnovne operacije na drevesu
 - Velikost problema je dolžina niza – k
 - Vstavljanje, brisanje, iskanje – $\mathcal{O}(k)$
- Urejanje
 - Velikost problema sta število besed in dolžina najdaljše besede – n in l
 - $\mathcal{O}(n \cdot l)$
- “Autocomplete”
 - Velikost problema so število besed, dolžina najdaljše besede in dolžina predpone – n , l in d
 - $\mathcal{O}(n(l - d))$

- Trie, Wikipedia (<https://en.wikipedia.org/wiki/Trie>)
(Datum dostopa: 12. 12. 2021)
- Sorting array strings words using trie, Geeks For Geeks
(<https://www.geeksforgeeks.org/sorting-array-strings-words-using-trie/>)
(Datum dostopa: 12. 12. 2021)
- Autocomplete feature using trie, Geeks For Geeks
(<https://www.geeksforgeeks.org/auto-complete-feature-using-trie/>)
(Datum dostopa: 12. 12. 2021)
- Trie, Interview cake
(<https://www.interviewcake.com/concept/java/trie>)
(Datum dostopa: 12. 12. 2021)