PSA

Teoretična 02 notes

PATRICIA TREE

Patricia trees ali practical algorithm to retrieve information in alpha-numeric so kompresirana različica binarnih radix dreves (prefix trees).

BASIC CONCEPTS

Tip trie drevesa, kjer so pogosti prefixi shranjeni samo 1x, da se izboljša space complexity. Trie drevesa imajo v nodes shranjeno po en znak, medtem ko imajo Patricijina drevesa združene single-child nodes.

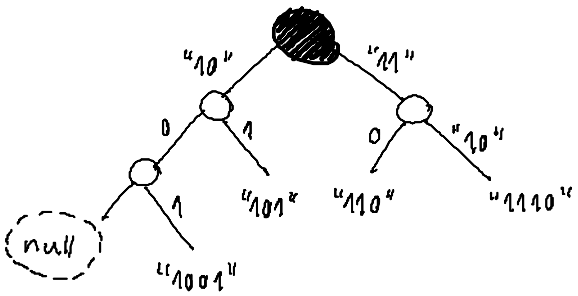
Nodes: vsak node lahko predstavlja več znakov (ali bitov)

Edges: (branches) med nodes so označeni z bitnimi indices, ki povejo, kateri bit med ključi naj pogleda naprej.

Compression: zaporedni nodes z enim otrokom so združena v eno vozlišče.

PRIMER DREVESA

101
1001
110
1110



(A je to prav...)

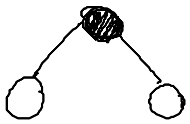
NALOGA 1: Patricijina drevesa. Predpostavimo, da smo vstavili $n \geq 2$ elementov v Patricijino drevo nad abecedo Σ in da noben element ni predpona nekega drugega.

(i) Če je $\Sigma = \{0,1\}$, pokaži, da ima Patricijino drevo natanko $n-1$ notranjih vozlišč.

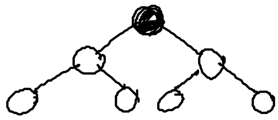
- Vsako vozlišče predstavlja en znak iz abecede.
- Robovi povezujejo vozlišča, ki predstavljajo zaporedne znake v nizu.
- Vsak list predstavlja konec nekega niza.
- Ni notranjih vozlišč z enim samim otrokom.

Vsako notranje vozlišče predstavlja razcep med dvema različnima nizoma.

1. Vsak list predstavlja en niz.
2. Vsako notranje vozlišče predstavlja razcep med dvema RAZLIČNIMA nizoma.
3. Če imamo n nizov, imamo n listov (po 1.)
4. Vsak list je dosegljiv po ENI SAMI POTI od korena.
5. Obstaja n poti (po 4).
6. Vsaka pot od korena do lista vsebuje natanko eno manjše število notranjih vozlišč, kot je število robov.
7. Če imamo n listov, imamo n-1 notranjih vozlišč



$n \geq 2$ oz. $n = 2 \Rightarrow$ notranja vozlišča $= n - 1 = 2 - 1 = 1$, v tem primeru je to koren drevesa.

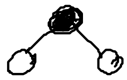


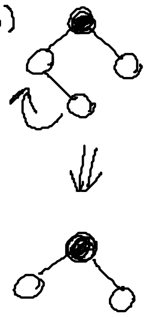
$n = 4 \geq 2 \Rightarrow$ notranja vozlišča $= 4 - 1 = 3$ $\square$

Notranje vozlišče je vsako vozlišče, ki ima vsaj enega otroka $\Leftrightarrow$ vsako vozlišče, ki ni list.

Elementi drevesa so njegovi LISTI ; zato je $n = 4$!

- Abeceda $\{0,1\}$ ima $|\{0,1\}| = 2$, zato nastane 1 razcep poti na vozlišče.
- $n \geq 2$ elementov zagotovi, da je vsaj koren z dvema otrokoma.

a)  $n = 2 \Rightarrow n.v. = 1$ $\square$

b) 

$n = 2 \Rightarrow n.v. = 1$, kar je v protislovju, TODA drevo ne dopušča vozlišč z enim otrokom, zato se združita onedva na levi strani! ;P

$n = 2 \Rightarrow n.v. = 1$ $\square$

Za binarna drevesa, ki imajo vedno 2 roba, je višina definirana kot

$$h = \log_2 n = \log n$$

Patricijina drevesa (oz. trie drevesa) pa imajo neko abecedo Σ, ki odloča o tem koliko robov bo obstajalo. Ker ima naša abeceda $|\Sigma| = 2$, to pomeni, da je višina:

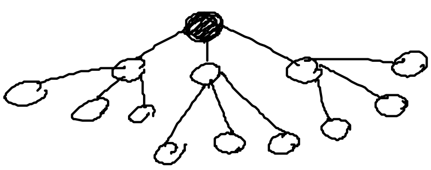
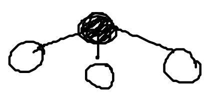
$$h = \log_{|\Sigma|} n = \log_2 n$$

Če ima drevo $n \geq 2$ notranjih vozlišč in $|\Sigma| = 2$, potem je njegova višina lahko:

$h = \log_2 2 = 1$, ali če je $n > 2$: $h = \log_2 n$, oziroma $n = 2^h$

(ii) Če je $\Sigma = \{0, 1, X\}$, kakšna je največja možna višina v Patricijinem drevesu in kakšna najmanjša?

Največja višina je takrat, ko drevo ne kompresira poti / vozlišč $\Leftrightarrow$ ko so vsi nizi med sabo različni.
Ker je $|\Sigma| = 3$, pride do treh vej/poddreves na vozlišče.

 $\Rightarrow n = 9$  $\Rightarrow n = 3$

$3^2 = 9$ $3^1 = 3$

$2 = \log_3 9 = h = 1 = \log_3 3$

!

$n = 9$ in $|\Sigma| = 3$

$\Downarrow$

$\log_3 9 = \log_3 3^2 = 2 \cdot \log_3 3 = 2 \cdot 1 = \underline{\underline{2}}$

Največja možna višina z $n \geq 3$ elementi (listi) in $\Sigma = \{0, 1, X\}$ in $|\Sigma| = 3$, je torej $\log_{|\Sigma|} n = \log_3 n$.

Najmanjša možna višina je, ko je $n = 3$ (in ne 2, saj mora imeti vsak node najmanj $|\Sigma|$ otroke - 3): $\log_3 3 = 1$.

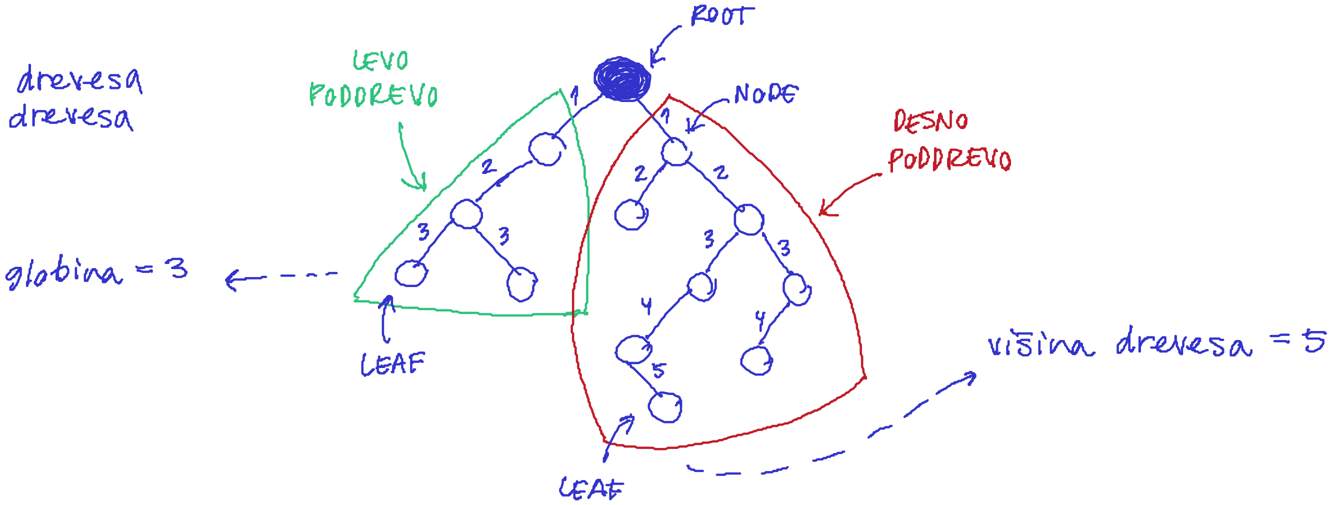
BINARY TREE

Podatkovna struktura, kjer ima vsak node največ DVA otroka (levi in desni) oz. poddrevesi.

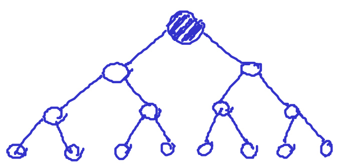
KEY PROPERTIES

Nodes – element drevesa
Root – najvišji (prvi) element brez starša
Child – node povezan na drugi node od zgoraj
Parent – obratno od child
Leaf – node brez otrok
Height – število razvejitev drevesa
Depth – število razvejitev do nekega node
Balanced tree – drevo, kjer je razlika v višini med levim in desnim poddrevesom katerekoli node minimizirana

AVL drevesa
RB drevesa



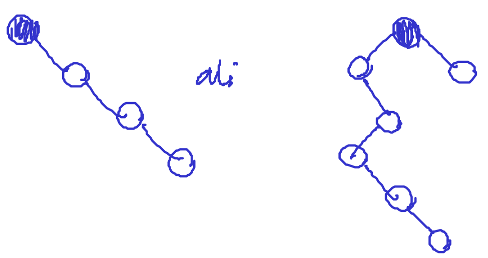
Primer uravnoteženega dvojiškega drevesa:



BINARY SEARCH TREE

Vse vrednosti v levem poddrevesu so manjše od root, medtem ko so vse v desnem večje.

Primer neuravnoteženega dvojiškega drevesa



ČASOVNA ZAHTEVNOST BT

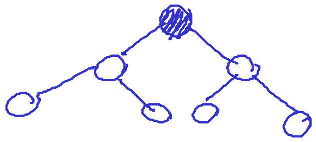
Odvisna je od višine dvojiškega drevesa. Naj bo višina označena s h. h je max število razdvajitev drevesa od root do nekega leaf. Višina odloča o tem, koliko primerjav je potrebnih v najslabšem primeru za insert, search, delete.

OPERACIJA	BEST-CASE	AVERAGE-CASE	WORST-CASE
Search	O(log n)	O(log n)	O(n)
Insert	O(log n)	O(log n)	O(n)
Delete	O(log n)	O(log n)	O(n)

ko je drevo neuravnoteženo!

Naj bo n število nodes v drevesu in h višina drevesa. V uravnovešenem dvojiškem drevesu je $h \doteq \log n$, kar vodi v $O(\log n)$ za večino operacij. V najslabšem primeru (neuravnovešeno, "degenerate" tree), je $h = n-1$, kar vodi v $O(n)$ za operacije.

↳ V bistvu pride do povezanega seznama, kar uniči smisel drevesa.

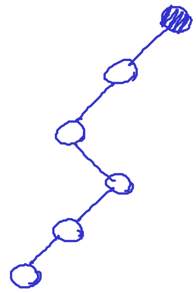


$n = 7$

$h = \log_2 7 \approx \lfloor 2.8074 \rfloor = 2$

$n = 6$

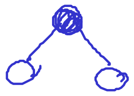
$h = 6 - 1 = 5$



NALOGA 2: Dvojiška drevesa. Višino drevesa definiramo kot število vozlišč, vključno s korenom, na najdaljši poti od korena do lista.

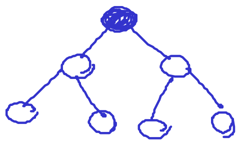
(i) Koliko največ vozlišč ima lahko dvojiško drevo višine h?

Vem, da je $h = \lfloor \log_2(n) \rfloor$, kar bi torej pomenilo $n = 2^h$?



$h = 1$

$n = \text{ROOT} + 2(\text{CHILD})$
$= 3$



$h = 2$

$n = \text{ROOT} + 2(\text{CHLD}) + 2(2(\text{CHILD}))$
$= \text{ROOT} + 2(\text{CHILD}) + 4(\text{CHILD})$
$= \text{ROOT} + 6(\text{CHILD})$
$= 7$