Ψηφιακό Δέντρο (Trie) και Δέντρο Επιθεμάτων (Suffix tree)

Υβριδικές Δομές Δεδομένων

Συνδυάζουν τη χρήση **δεικτών** και **πινάκων**

- Ψηφιακό δέντρο (Trie) και Δέντρο επιθεμάτων (Suffix tree)
- Interpolation Search Tree

Ψηφιακό δέντρο (Trie)

- Αποθήκευση και ανάκτηση πληροφορίας κειμένου εύκολα και γρήγορα
  - Λέξεις
  - Συμβολοσειρές
  - Επιθέματα
  - Κτλ.
- Ταίριασμα Προτύπου ή Συμβολοσειράς
  - Εφαρμογές σε αναζήτηση/επεξεργασία κειμένου, data mining, βιοπληροφορική κτλ.

Λογική του Trie

- Έστω $S = \{x_1, \dots, x_n\}$
- Θέλω να αναπαραστήσω το S σε μια δομή
- Δεν στηριζόμαστε στις τιμές x_i
- Χρησιμοποιούμε αναπαράσταση των στοιχείων ως μία ακολουθία χαρακτήρων
- Οι λέξεις βρίσκονται ανάλογα με το γράμμα με το οποίο αρχίζουν (το ίδιο και με τα υπόλοιπα γράμματα) → μοιάζει με λεξικό
- Είναι **φυλλοπροσανατολισμένο**

Ορισμός

- Έστω σύμπαν U του οποίου τα στοιχεία είναι συμβολοσειρές μήκους λ πάνω σε ένα αλφάβητο K με $|K| = k$. Ένα σύνολο $S \subseteq U$ αναπαρίσταται ως ένα k-δικό δέντρο που περιέχει όλα τα προθέματα των στοιχείων του S

Υλοποίηση

1. Κάθε εσωτερικός κόμβος του δέντρου είναι ένας πίνακας μήκους k από δείκτες
2. Κάθε θέση του πίνακα αντιστοιχίζεται σε ένα γράμμα του αλφαβήτου
3. Κάθε θέση του πίνακα σε ένα κόμβο υ σε βάθος i θα λάβει τιμή αν κάποιο από τα στοιχεία του S στην i-οστή θέση έχει τον αντίστοιχο χαρακτήρα
4. Ύψος λ
5. Χώρος O(k)

INSERT(x)

1. $u \leftarrow$ρίζα του $TRIE$
2. **for**$(i = 0; i < \lambda; i++)$ {
3. **if**($x[i]$-οστό παιδί του $u = null$)(* δηλ. δεν υπάρχει *)
4. **then** δημιούργησε το $x[i]$′-οστό παιδί
5. $u \leftarrow x[i]$-οστό παιδί του u
   (*$x[i]$ είναι ο i-οστό ψηφίο από αριστερά του στοιχείου x
   Το $x[i]$-οστό παιδί υποδεικνύεται από τον δείκτη της θέσης του πίνακα
   του κόμβου που έχει αντιστοιχισθεί στο γράμμα του αλφαβήτου
   που συναντάμε στην θέση $x[i]$ *)
6. }
7. περιεχόμενο(u) $\leftarrow x$

ACCESS(x)

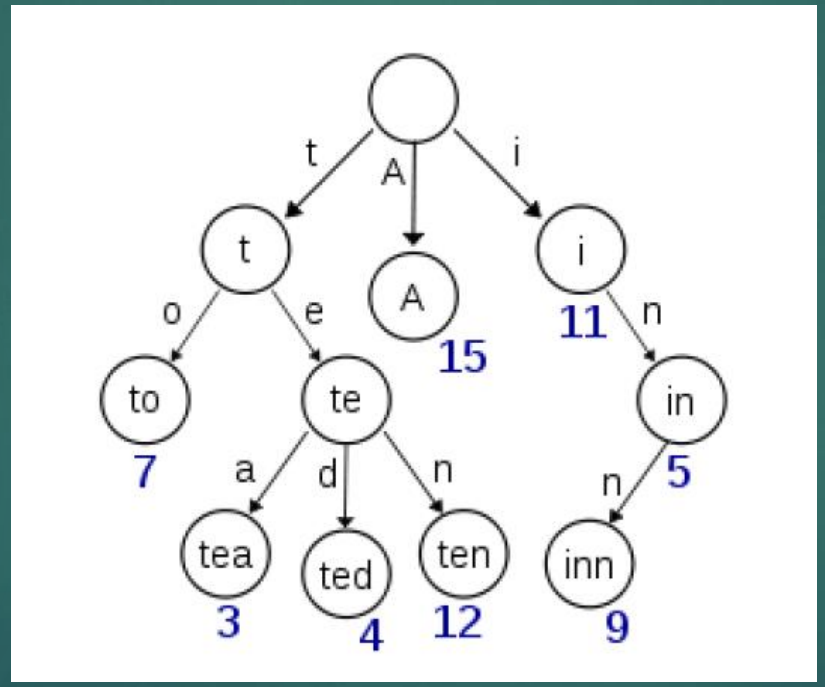
1. $u \leftarrow$ρίζα του $TRIE$
2. **for**$(i = 0; i < \lambda; i++)$ {

3. $u \leftarrow x[i]$-οστό παιδί του u
   (*$x[i]$ είναι ο i-οστό ψηφίο από αριστερά του στοιχείου x
   Το $x[i]$-οστό παιδί υποδεικνύεται από τον δείκτη της θέσης του πίνακα
   του κόμβου που έχέι αντιστοιχισθεί στο γράμμα του αλφαβήτου
   που συναντάμε στην θέση $x[i]$ *)
4. }
5. **if**$(x =$ περιεχόμενο(u)
6. **then** Επιτυχία: $x \in S$
7. **else** Αποτυχία: $x \notin S$

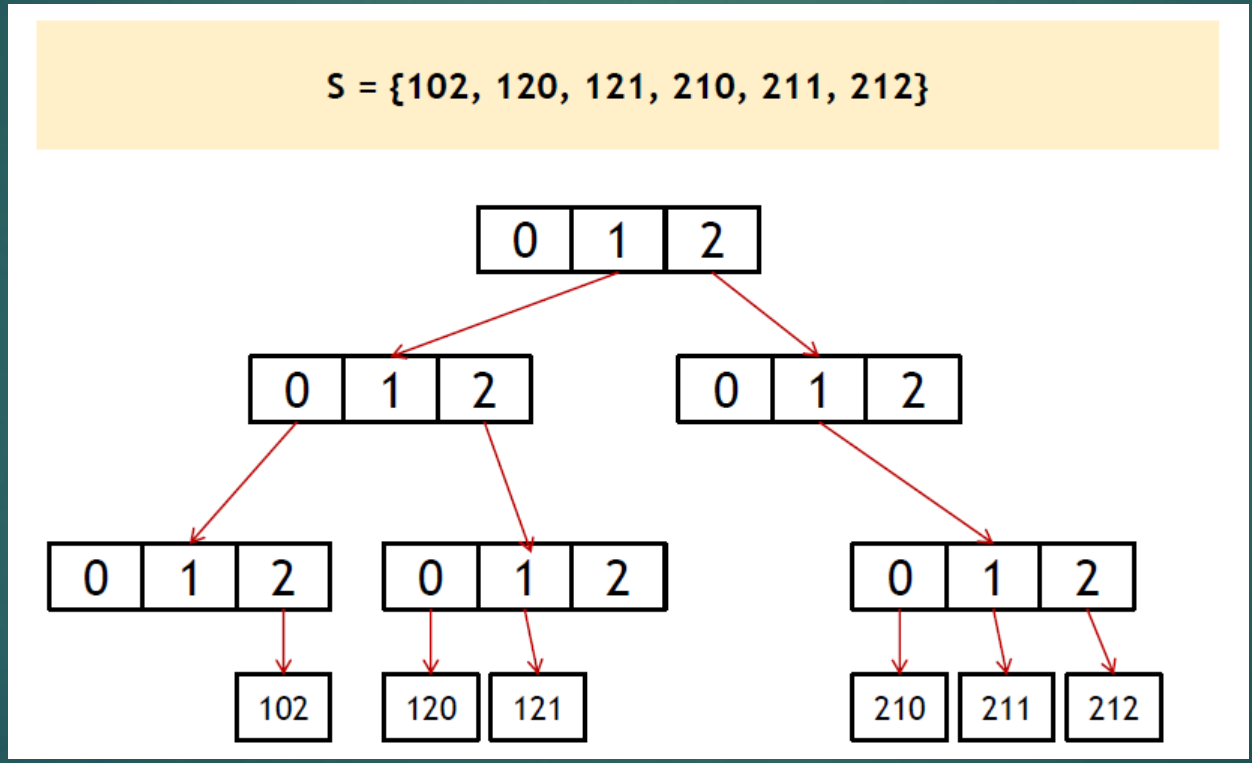
DELETE(x)

1. $Access(x)$ κρατώντας σε μία στοίβα τους κόμβους
   καθώς κατεβαίνουμε το δένδρο
2. $u \leftarrow$ κορυφή της στοίβας
3. **if**($x =$ περιεχόμενο(u)) (*δηλ $x \in S$*)
4. **then**{
5. διαγραφή του u
6. διαγραφή της κορυφής της στοίβας μέχρι αυτή να είναι κόμβος με παιδί
7. }

Παράδειγμα 1



Παράδειγμα 2



Πολυπλοκότητες

- Οι βασικές πράξεις απαιτούν χρόνο $O(\lambda) = O(\ log_k N)$
  - Εξαρτώνται από το μέγεθος του αλφαβήτου k και του σύμπαντος U

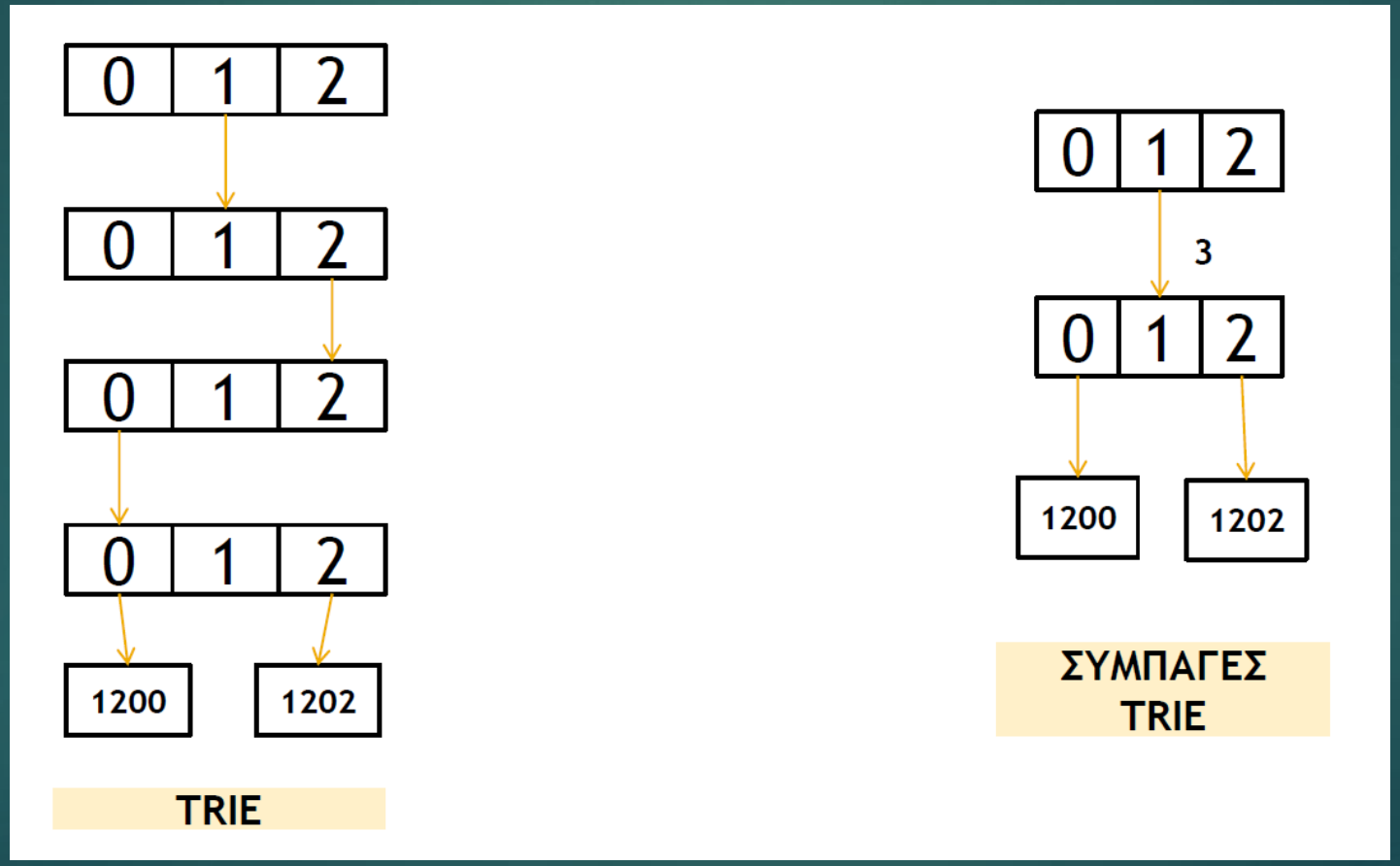
- Ο χώρος που απαιτείται στην χειρότερη περίπτωση είναι $O(n\lambda k),\ n = |S|$
  Αυτό συμβαίνει όταν τα στοιχεία δεν έχουν κοινά προθέματα:
  - n πλήρη μονοπάτια
  - $n\lambda$ κόμβοι συνολικά
  - $n\lambda k$ συνολικός χώρος

Συμπαγές Trie

- Αποθηκεύονται μόνο οι κόμβοι του Trie με βαθμό μεγαλύτερο ή ίσο του 2, ενώ οι αλυσίδες των κόμβων με βαθμό 1 αντικαθίστανται από ένα απλό αριθμό
- Ο αριθμός αυτός αποθηκεύεται στην (πρώτη) πλευρά που οδηγεί στην αλυσίδα και είναι ίσος με το πλήθος των κόμβων σε αυτή
- Στην χειρότερη περίπτωση, ο χώρος από $O(n\lambda k)$ μειώνεται σε $O(nk)$

Θεώρημα 6.2. *Το αναμενόμενο ύψος H ενός συμπαγούς $TRIE$ n στοιχείων είναι $O(\log_k n)$.*

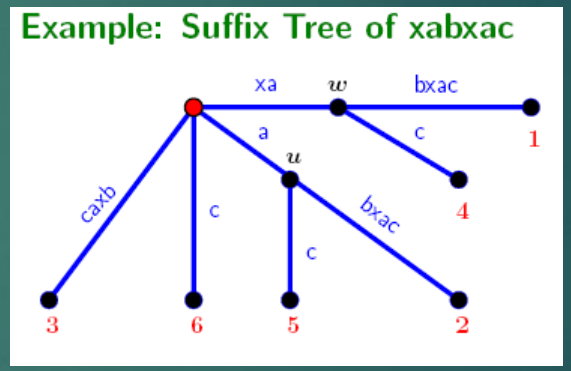
Παράδειγμα


0 1 2
0 1 2
0 1 2
0 1 2
1200
1202
TRIE
0 1 2
3
0 1 2
1200
1202
ΣΥΜΠΑΓΕΣ
TRIE

Δέντρο Επιθεμάτων (Suffix Tree)

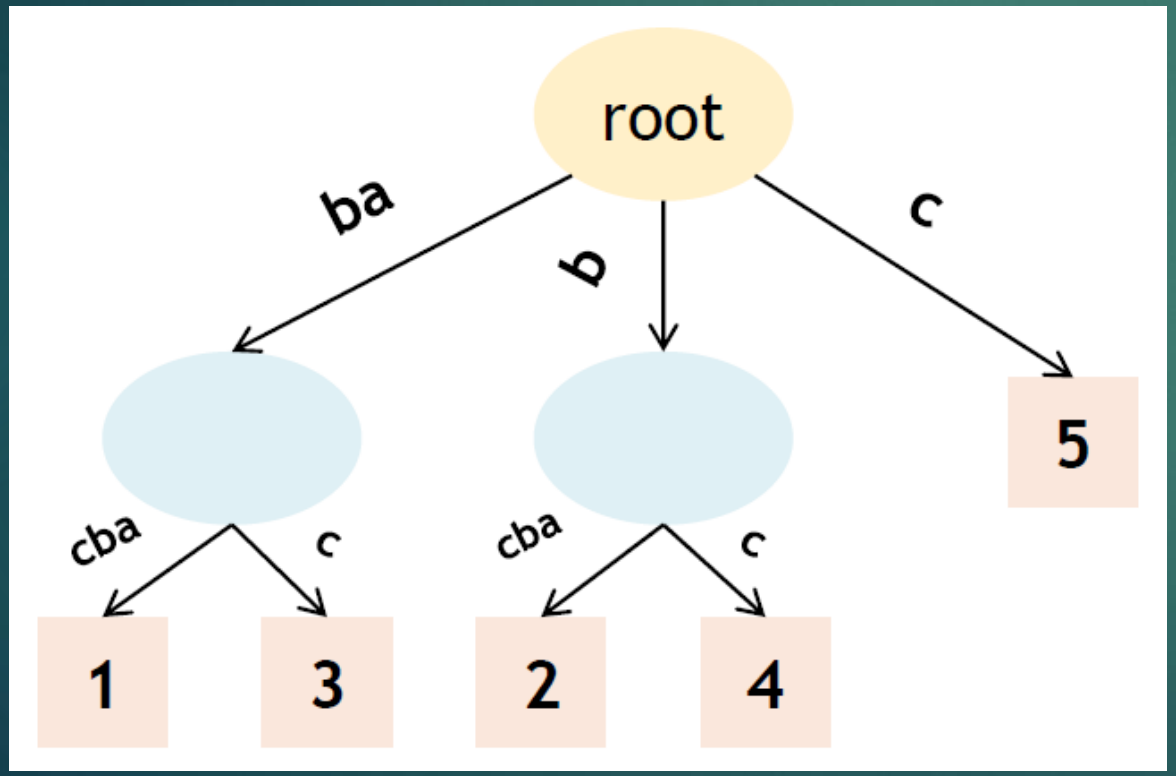
- Αποθηκεύει όλα τα δυνατά επιθέματα μιας συμβολοσειράς S.

- **Ορισμός**
  Το Suffix Tree μιας συμβολοσειράς $S[1 \dots n]$ είναι ένα συμπαγές Trie που περιέχει ως κλειδιά όλα τα επιθέματα $S[1 \dots n]$, $\mathbf{1 \leq i \leq n}$



Κατασκευή Suffix Tree

Έστω συμβολοσειρά X=ababc

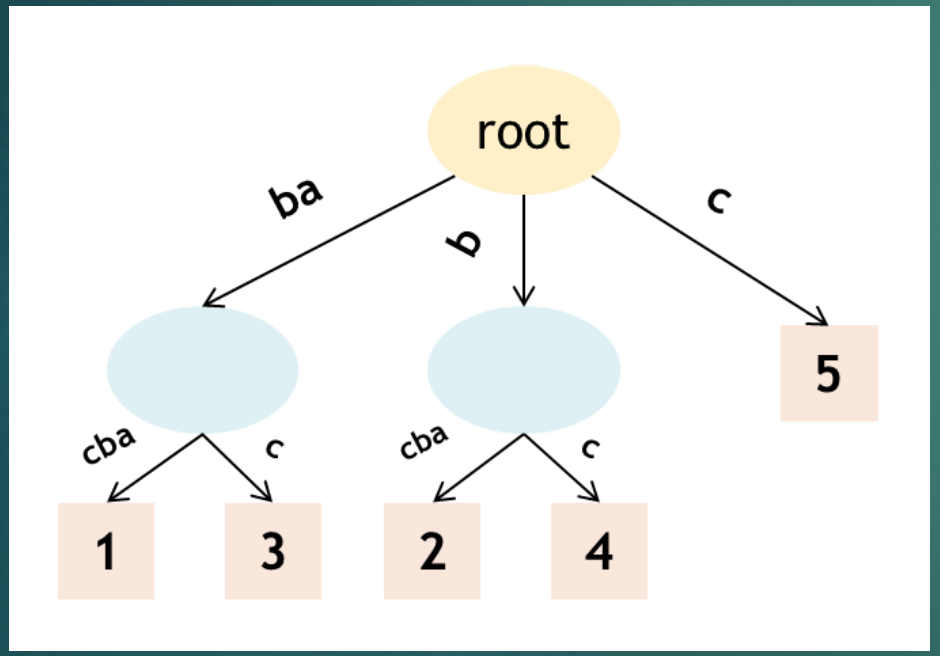


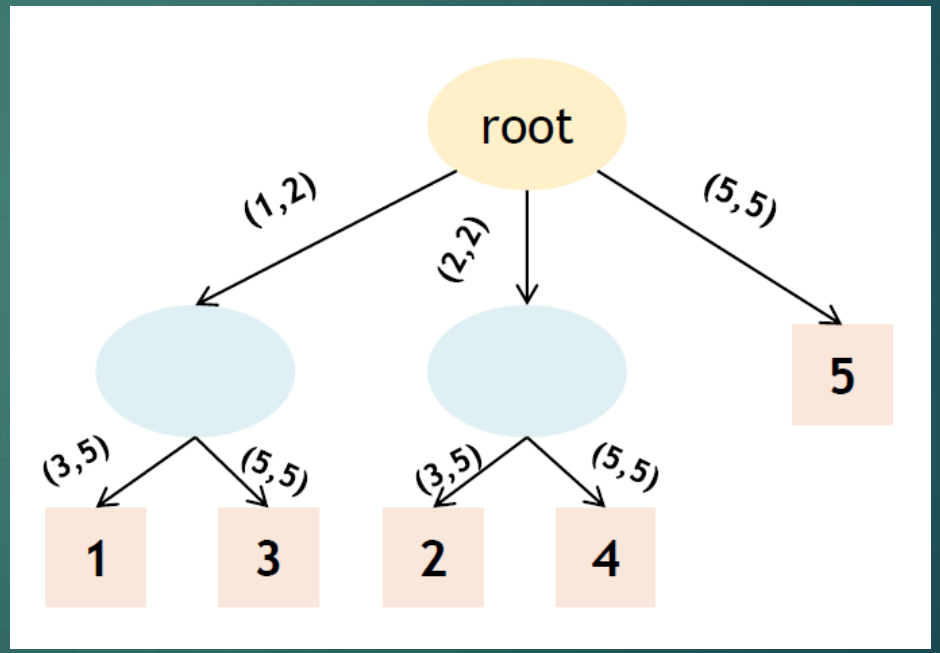
ΠΡΟΣΟΧΗ!

Από τον ίδιο κόμβο δεν εξέρχονται υποσυμβολοσειρές με κοινό πρώτο χαρακτήρα

Κατασκευή Suffix Tree – Αλλιώς

Έστω συμβολοσειρά X=ababc





Ορισμός 6.3. *Ένα δένδρο επιθεμάτων T για μια συμβολοσειρά X μεγέθους n ($|X| = n$) είναι μία κατευθυνόμενη δενδρική δομή με ακριβώς n φύλλα τα οποία είναι αριθμημένα από το 1 μέχρι το n. Κάθε εσωτερικός κόμβος, ο οποίος δεν είναι η ρίζα, έχει τουλάχιστον δύο παιδιά και κάθε πλευρά αντιστοιχίζεται σε μία μη-μηδενική υπο-συμβολοσειρά της X. Οι υπο-συμβολοσειρές των πλευρών που εξέρχονται από τον ίδιο κόμβο δεν επιτρέπεται να έχουν κοινό τον πρώτο τους χαρακτήρα. Τέλος κύριο χαρακτηριστικό του δένδρου επιθεμάτων είναι το γεγονός ότι αν ενώσουμε τις υπο-συμβολοσειρές που συναντάμε πηγαίνοντας από την ρίζα σε κάποιο από τα φύλλα, έστω το φύλλο με αριθμό i, σχηματίζεται το επίθεμα της συμβολοσειράς X που ξεκινά από την θέση i, δηλαδή το $X[i..n]$.*

Εφαρμογές Suffix Tree

1. Ταίριασμα Προτύπου – Pattern matching
2. Μέγιστη Επαναλαμβανόμενη Υποσυμβολοσειρά – Longest Repeated Substring
3. Μέγιστη Κοινή Υποσυμβολοσειρά – Longest Common Substring