Δομές Δεδομένων

Συμβολοσειρές

Δημήτρης Μιχαήλ

Τμήμα Πληροφορικής και Τηλεματικής
Χαροκόπειο Πανεπιστήμιο

Συμβολοσειρές

Συμβολοσειρές και προβλήματα που αφορούν συμβολοσειρές εμφανίζονται τόσο συχνά που αξίζουν ξεχωριστή μελέτη όσο αναφορά τις δομές δεδομένων.

Συμβολοσειρές

Μία συμβολοσειρά είναι ουσιαστικά ένας πίνακας ακεραίων.

Συνήθως όμως επιλέγουμε να διαχωρίζουμε μεταξύ ενός

- πίνακα ακεραίων και μιας
- συμβολοσειράς χαρακτήρων από ένα αλφάβητο.

Αλφάβητο

Έστω Σ ένα γνησίως διατεταγμένο (συνήθως πεπερασμένο) σύνολο, που ονομάζεται το **αλφάβητο**.

- Τα στοιχεία του Σ ονομάζονται χαρακτήρες.
- Για δύο χαρακτήρες $a \in \Sigma$ και $b \in \Sigma$ γράφουμε $\alpha < \beta$ για να συμβολίσουμε πως ο χαρακτήρας α είναι μικρότερος από τον β.

Συμβολοσειρές

Μία συμβολοσειρά S από το Σ ορίζεται ως μία ακολουθία χαρακτήρων $a \in \Sigma$.

- Γράφουμε $|S|$ για να συμβολίσουμε το μήκος της S (αριθμό χαρακτήρων).
- Συμβολίζουμε με S_i τον i-οστό χαρακτήρα της S.
- Γράφουμε S_{ij} για την υποσυμβολοσειρά της S από τον χαρακτήρα i έως και τον χαρακτήρα j.

Εάν S, T είναι δύο συμβολοσειρές, συμβολίζουμε με ST την παράθεση τους.

Σύγκριση Συμβολοσειρών

Έστω $a, b \in \Sigma$ και S, T δύο συμβολοσειρές πάνω στο Σ.

Γράφουμε

$$aS < bT$$

ή

”η συμβολοσειρά aS είναι **λεξικογραφικά** μικρότερη της bT”

εάν $a < b$ ή $a = b$ και $S < T$.

- Η κενή λέξη ϵ θεωρείται μικρότερη από οποιαδήποτε μη-κενή συμβολοσειρά.
- Επίσης γράφουμε $S \leq T$ για να συμβολίσουμε πως $S < T$ ή $S = T$.

Πρόθημα, επίθημα, υποσυμβολοσειρές

πρόθημα (prefix)

Ένα **πρόθημα** μιας συμβολοσειράς S προκύπτει αφαιρώντας μηδέν ή περισσότερους χαρακτήρες από το τέλος της S.

π.χ η συμβολοσειρά ban είναι πρόθημα της banana

επίθημα (suffix)

Ένα **επίθημα** μιας συμβολοσειράς S προκύπτει αφαιρώντας μηδέν ή περισσότερους χαρακτήρες από την αρχή της S.

π.χ η συμβολοσειρά nana είναι επίθημα της banana

Πρόθημα, επίθημα, υποσυμβολοσειρές

υποσυμβολοσειρά (substring)

Μια **υποσυμβολοσειρά** μιας συμβολοσειράς S προκύπτει αφαιρώντας ένα πρόθημα και ένα επίθημα από την S.

γνήσιο πρόθημα, επίθημα ή υποσυμβολοσειρά

Έστω X ένα πρόθημα, επίθημα ή υποσυμβολοσειρά μιας συμβολοσειράς S. Ονομάζετε γνήσιο(α) σε περίπτωση που

- $X \neq \epsilon$ και
- $X \neq S$.

Υποακολουθία

υποακολουθία (subsequence)

Οποιαδήποτε συμβολοσειρά προκύπτει σβήνοντας μηδέν ή περισσότερα σύμβολα, όχι υποχρεωτικά συνεχόμενα, από την συμβολοσειρά S.

π.χ η συμβολοσειρά baaa είναι υποακολουθία της banana

Ταίριασμα Συμβολοσειράς

String-Matching Problem

Το πρόβλημα Ταιριάσματος Προτύπου ή Συμβολοσειράς (Pattern-Matching ή String-Matching Problem) αφορά τον εντοπισμό όλων των εμφανίσεων μίας δεδομένης συμβολοσειράς-λέξης-προτύπου σε μία άλλη συμβολοσειρά-κείμενο συνήθως μεγαλύτερου μήκους.

Ταίριασμα Συμβολοσειράς

String-Matching Problem

Το πρόβλημα Ταιριάσματος Προτύπου ή Συμβολοσειράς (Pattern-Matching ή String-Matching Problem) αφορά τον εντοπισμό όλων των εμφανίσεων μίας δεδομένης συμβολοσειράς-λέξης-προτύπου σε μία άλλη συμβολοσειρά-κείμενο συνήθως μεγαλύτερου μήκους.

Χωρίζεται σε δύο βασικές κατηγορίες, το *ακριβές* και το *προσεγγιστικό* ταίριασμα.

Ακριβές Ταίριασμα Συμβολοσειράς

Δεδομένου μίας συμβολοσειράς κειμένου T μήκους n και μίας συμβολοσειράς προτύπου P μήκους $m \leq n$, βρείτε όλες τις εμφανίσεις του προτύπου P στο κείμενο T.

Ακριβές Ταίριασμα Συμβολοσειράς

Δεδομένου μίας συμβολοσειράς κειμένου T μήκους n και μίας συμβολοσειράς προτύπου P μήκους $m \leq n$, βρείτε όλες τις εμφανίσεις του προτύπου P στο κείμενο T.

- Η προφανής λύση αναζήτησης πέρνει $\mathcal{O}(mn)$ συγκρίσεις αφού αναζητά όλο το πρότυπο για κάθε θέση του κειμένου.
- Πιο προχωρημένες τεχνικές λύνουν το πρόβλημα σε $\mathcal{O}(m+n)$ συγκρίσεις στην χειρότερη περίπτωση που είναι το βέλτιστο.

Στατικό Κείμενο - Πολλά Ερωτήματα

Πολλές φορές το κείμενο είναι *στατικό* και το γνωρίζουμε εξαρχής και πρέπει να απαντήσουμε σε πολλά ερωτήματα που εμφανίζονται ένα ένα.

Ευρετήριο

Σε αυτή την περίπτωση έχει νόημα να ξοδέψουμε χρόνο ώστε να χτίσουμε κάποιο ευρετήριο του κειμένου T ώστε να απαντάμε τα ερωτήματα αποδοτικά.

Βασικές Δομές Δεδομένων για Συμβολοσειρές

- Trie (prefix tree)
- Patricia Trie (radix tree)
- Suffix Tree
- Suffix Array

Trie

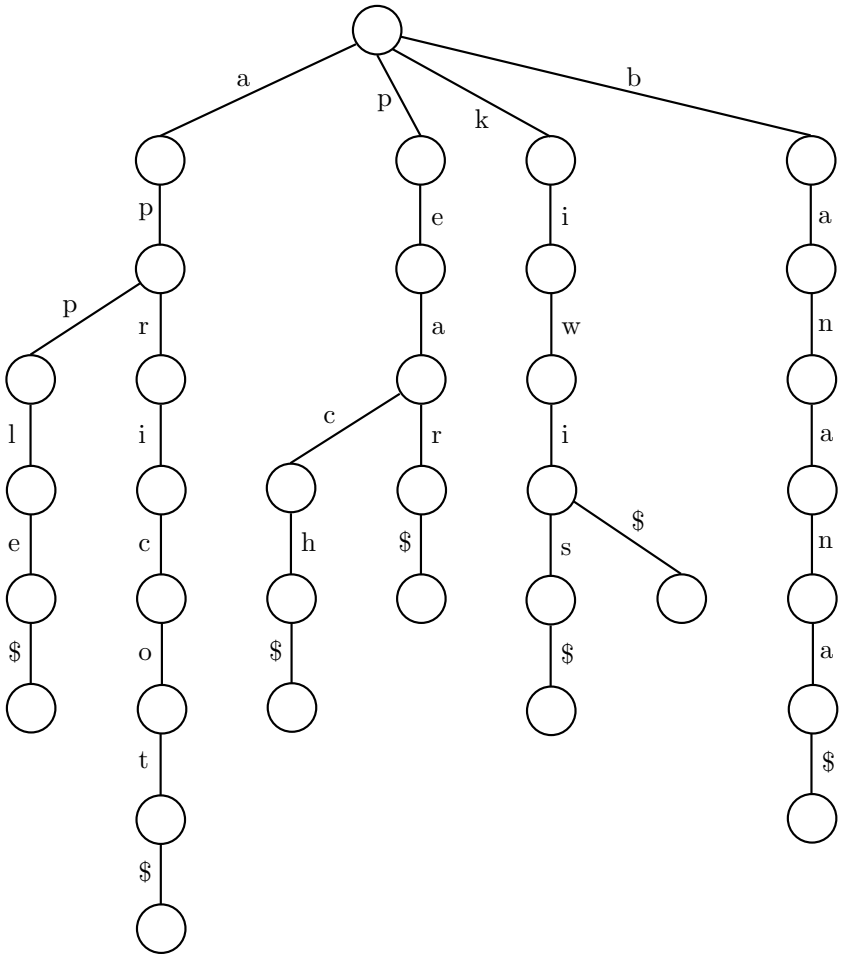
Retrieval

Ένα δέντρο αναζήτησης σχεδιασμένο αποκλειστικά για κλειδιά συμβολοσειρές.

E. Fredkin. Trie Memory. Comm. ACM 3(9) pp490-499 Sept. 1960.

Trie

Retrieval

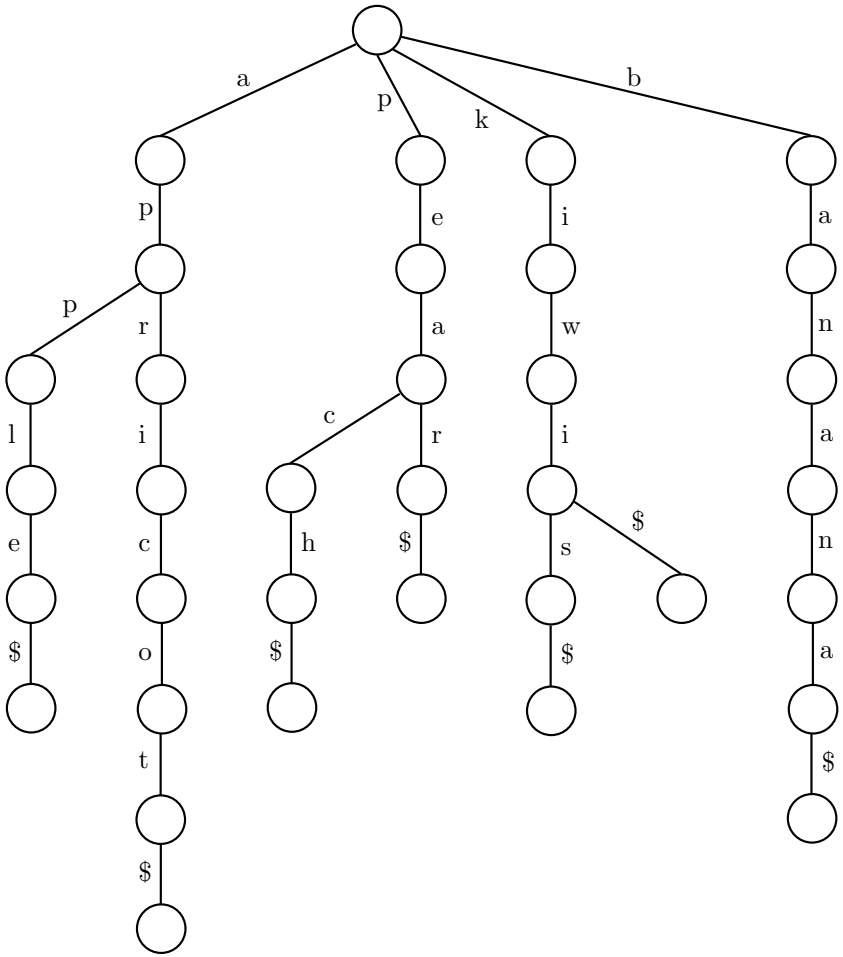


Παράδειγμα με κλειδιά

- apple
- apricot
- peach
- pear
- kiwi
- kiwis
- banana

Trie

Retrieval

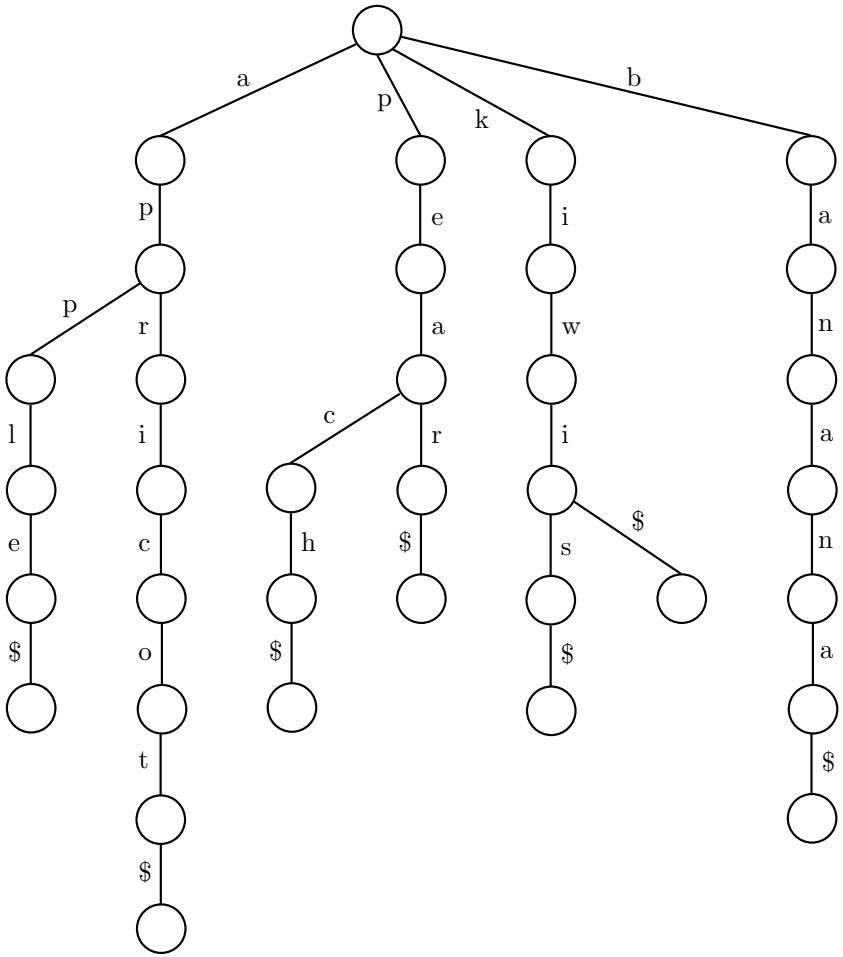


Ιδιότητες

- Κάθε κόμβος έχει το πολύ $|\Sigma|$ παιδιά.
- Κάθε ακμή έχει ως ετικέτα έναν χαρακτήρα $\in \Sigma$.
- Κάθε κόμβος αντιστοιχεί με ένα κλειδί, την παράθεση των χαρακτήρων στο μονοπάτι από την ρίζα στον κόμβο.

Trie

Retrieval



Ιδιότητες

- Η ρίζα αντιστοιχεί στην κενή συμβολοσειρά.
- Όλοι οι απόγονοι ενός κόμβου *v* έχουν κοινό πρόθημα το κλειδί του κόμβου *v*.
- Οι τιμές που αντιστοιχούν στα κλειδιά αποθηκεύονται επάνω στους κόμβους.

Trie

Αναζήτηση

```
1  tree Trie–Search(tree root, string P[k..m])
2  {
3      if (root is leaf)
4          return root;
5
6      tree child = root.child(P[k]);
7      if (child == null)
8          return null;
9
10     return Trie–Search(child, P[k+1..m])
11 }
```

Ο αλγόριθμος αναζήτησης απλά ακολουθεί το μονοπάτι στο δέντρο ξεκινώντας με Trie-Search(root, P[0..m]).

Εισαγωγή

- Ακολουθούμε το μονοπάτι μέχρι είτε να βρούμε το κλειδί ή να βρούμε null.
- Εαν βρούμε null δημιουργούμε ένα καινούριο παρακλάδι με το υπόλοιπο της συμβολοσειράς.

Διαγραφή

- Αναζητάμε την συμβολοσειρά και ακολουθούμε την ανάποδη διαδικασία.
- Διαγράφουμε κόμβους από το τέλος προς την ρίζα όσο δεν υπάρχει κάποιο άλλο παρακλάδι.

Ο κόμβος σε ένα Trie μπορεί να περιέχει μέχρι και $|\Sigma|$ παιδιά.

Ποια αναπαράσταση να χρησιμοποιήσουμε και πόσο χρόνο χρειάζεται η λειτουργία *t.child*(i);

- Πίνακας με $|\Sigma|$ δείκτες: χάσιμο χώρου, αλλά $\mathcal{O}(1)$ για child(c).
- Πίνακας κατακερματισμού: μικρότερο χάσιμο χώρου, child(c) σε αναμενόμενο $\mathcal{O}(1)$
- Λίστα από δείκτες: λίγος χώρος, αλλά $\mathcal{O}(|\Sigma|)$ για child(c)
- Ισοζυγισμένο δυαδικό δέντρο αναζήτησης: λίγος χώρος και $\mathcal{O}(\log |\Sigma|)$ για child(c)

Trie

Ανάλυση

- Μέγεθος:
  - $\mathcal{O}(N)$ στην χειρότερη περίπτωση όπου N ο συνολικός αριθμός χαρακτήρων των n συμβολοσειρών που είναι στο trie.
- Αναζήτηση, εισαγωγή και διαγραφή συμβολοσειράς μεγέθους m:
  - $\mathcal{O}(m|\Sigma|)$, $\mathcal{O}(m \log |\Sigma|)$ ή $\mathcal{O}(m)$ ανάλογα με είδος κόμβου.
  - Ένα ισοζυγισμένο δυαδικό δέντρο αναζήτησης με κλειδιά συμβολοσειρές θα ήθελε $\mathcal{O}(m \log n)$.
- Παρατήρηση:
  - Οι αλυσίδες κόμβων με ένα παιδί δεν είναι αποδοτικές.

Patricia Trie

Το όνομα Patricia προέρχεται από το ακρώνυμο PATRICIA:

”**P**ractical **A**lgorithm **T**o **R**etrieve **I**nformation **C**oded **I**n **A**lphanumeric”

Patricia Trie

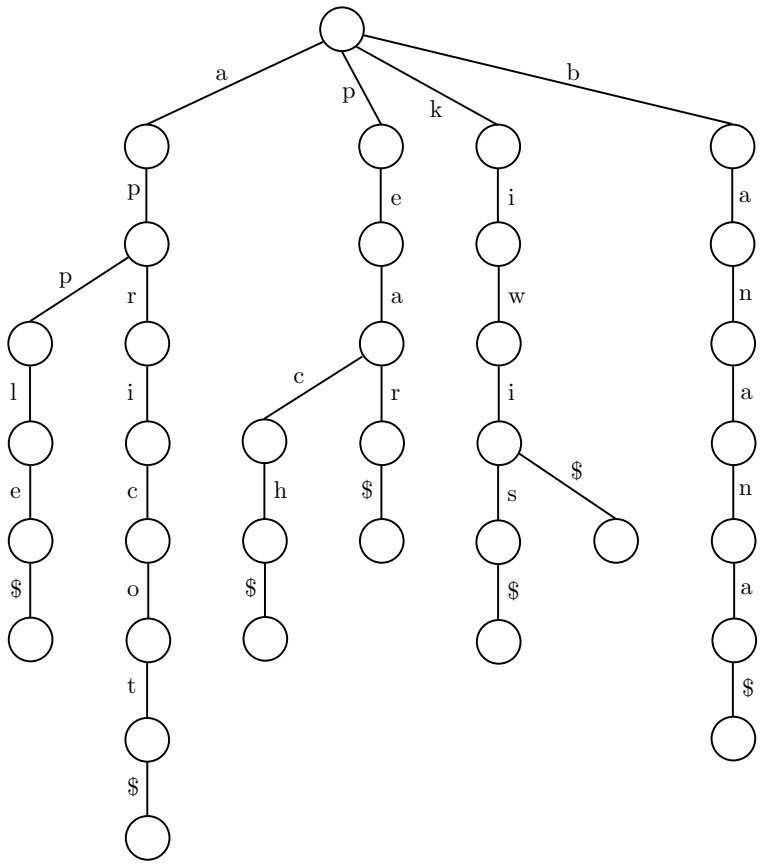
Το όνομα Patricia προέρχεται από το ακρώνυμο PATRICIA:

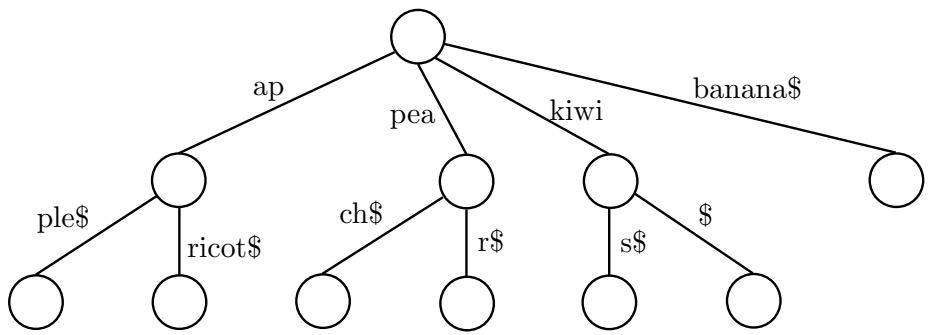
”**P**ractical **A**lgorithm **T**o **R**etrieve **I**nformation **C**oded **I**n **A**lphanumeric”

- Επιλύει το πρόβλημα των tries λόγω των κόμβων με ένα μοναδικό παιδί.
- Αντικαθιστούμε μία αλυσίδα από κόμβους με ένα παιδί, με μία ακμή που έχει συμβολοσειρά ως ετικέτα.

Patricia Trie

Οι ακμές μπορούν πλέον να περιέχουν συμβολοσειρές αντί μόνο έναν χαρακτήρα.





Κάθε μη-φύλλο εκτός από την ρίζα έχει τουλάχιστον δύο παιδιά.

Ταίριασμα Συμβολοσειράς

Πρόβλημα

Δεδομένου μίας συμβολοσειράς κειμένου T μήκους n και μίας συμβολοσειράς προτύπου P μήκους $m \leq n$, βρείτε όλες τις εμφανίσεις του προτύπου P στο κείμενο T.

```
T = "bananas"
P = "na"
```

Επαναδιατύπωση

Θέλουμε να βρούμε όλα τα επιθήματα (suffixes) του T που περιέχουν το πρότυπο P ως πρόθημα (prefix).

```
bananas$
 ananas$
  nanas$
   anas$
    nas$
     as$
      s$
       $
```

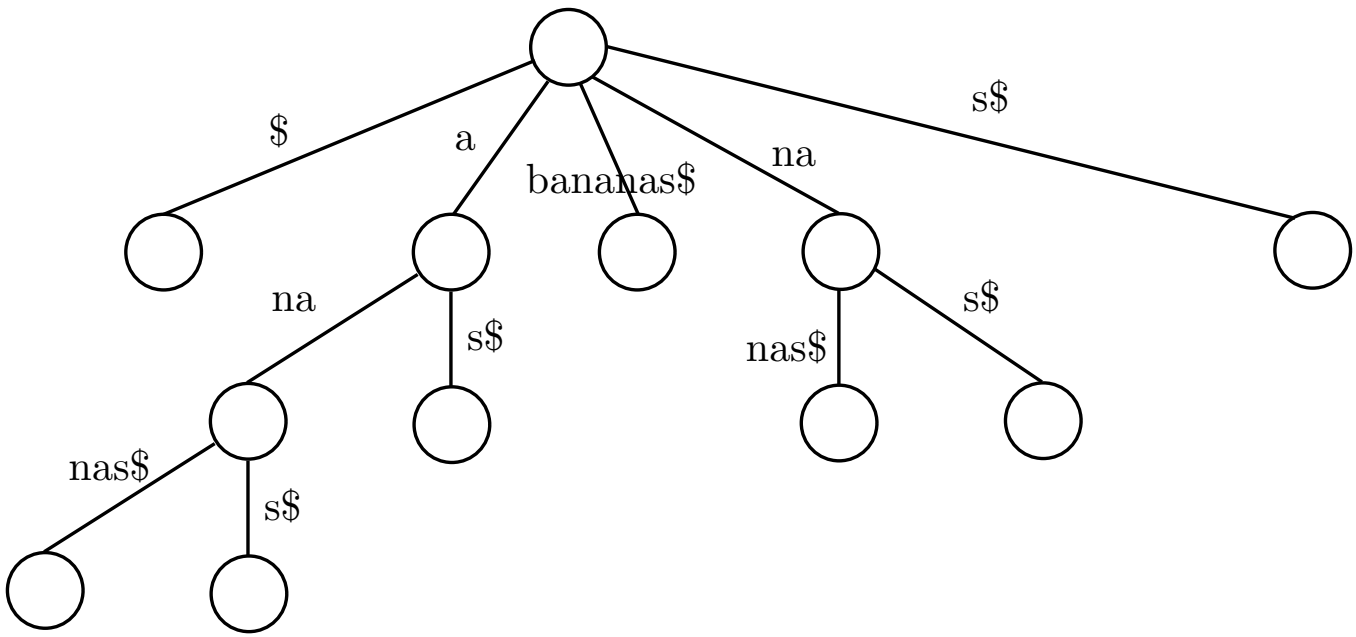
Επιθηματικά Δέντρα

Suffix Tree

Ένα επιθηματικό (Suffix) δέντρο για μία συμβολοσειρά T είναι ένα Patricia Trie που περιέχει όλα τα επιθήματα της T.

Επιθηματικά Δέντρα

Suffix Tree




```
bananas$
 ananas$
  nanas$
   anas$
    nas$
     as$
      s$
       $
```

Επιθηματικά Δέντρα

Suffix Tree

Ένα επιθηματικό δέντρο μίας συμβολοσειράς-κειμένου T μεγέθους n χαρακτήρων μπορεί να κατασκευαστεί σε χρόνο $\mathcal{O}(n)$ εάν το μέγεθος του αλφάβητου είναι πολυωνυμικό [Farach '1997].

Πόσο χρόνο πέρνει η αναζήτηση μίας συμβολοσειράς (προτύπο) P μεγέθους m χαρακτήρων εάν έχουμε ήδη κατασκευάσει ένα επιθηματικό δέντρο για την συμβολοσειρά (κείμενο) T μεγέθους n;

Πόσο χρόνο πέρνει η αναζήτηση μίας συμβολοσειράς (προτύπο) P μεγέθους m χαρακτήρων εάν έχουμε ήδη κατασκευάσει ένα επιθηματικό δέντρο για την συμβολοσειρά (κείμενο) T μεγέθους n;

- Η αναζήτηση είναι όπως στα Patricia Trees και πέρνει χρόνο $\mathcal{O}(m)$ ανεξάρτητο από το μέγεθος του κειμένου T.

Πολλές επιπλέον λειτουργίες μπορούν να γίνουν αποδοτικά με την χρήση επιθηματικών δέντρων.

Για περισσότερες πληροφορίες

- Gusfield, Dan: ”Algorithms on Strings, Trees and Sequences: Computer Science and Computational Biology”. USA: Cambridge University Press, 1997. ISBN 0-521-58519-8.

Επιθηματικός Πίνακας

Suffix Array

Ένας επιθηματικός πίνακας (suffix array) είναι ένας πίνακας ακεραίων με τις αρχικές θέσεις των επιθημάτων μίας συμβολοσειράς T ταξινομημένες κατά λεξικογραφική σειρά.

Επιθηματικός Πίνακας

Suffix Array

Η λέξη `bananas` έχει 8 επιθέματα μαζί με την κενή συμβολοσειρά

1	2	3	4	5	6	7	8
b	a	n	a	n	a	s	$

Επιθηματικός Πίνακας

Suffix Array

Η λέξη `bananas` έχει 8 επιθέματα μαζί με την κενή συμβολοσειρά

1	2	3	4	5	6	7	8
b	a	n	a	n	a	s	$

Κατασκευάζουμε έναν πίνακα που στην θέση *i* περιέχει την θέση που ξεκινάει το επίθημα που είναι *i*-οστό λεξικογραφικά.

```
        $
   ananas$
     anas$
       as$
 bananas$
    nanas$
      nas$
        s$
```

8	2	4	6	1	3	5	7

Επιθηματικός Πίνακας

Suffix Array

Ένας επιθηματικός πίνακας μπορεί να κατασκευαστεί σε χρόνο $\mathcal{O}(n)$ όπου n είναι ο αριθμός χαρακτήρων της συμβολοσειράς εισόδου.

Σε περίπτωση που υπάρχει ήδη ένα επιθηματικό δέντρο, μπορούμε να κατασκευάσουμε έναν επιθηματικό πίνακα με μία λεξιγραφική διάσχιση κατά βάθος, δηλαδή μία διάσχιση κατά βάθος όπου επισκεπτόμαστε τα παιδιά ενός κόμβου με λεξικογραφική σειρά.

Πρόβλημα

Θέλουμε να βρούμε όλα τα επιθήματα (suffixes) του T που περιέχουν το πρότυπο P ως πρόθημα (prefix).

Τα επιθήματα είναι ταξινομημένα λεξικογραφικά και άρα όλα τα επιθήματα που ψάχνουμε εμφανίζονται σειριακά.

Κάνουμε δύο δυαδικές αναζητήσεις για να βρούμε την μικρότερη θέση i του πίνακα όπου το P είναι πρόθημα του i-οστού επιθήματος και την μεγαλύτερη θέση j όπου το P είναι πρόθημα του j-οστού επιθήματος.

Πρόβλημα

Θέλουμε να βρούμε όλα τα επιθήματα (suffixes) του T που περιέχουν το πρότυπο P ως πρόθημα (prefix).

Κάθε δυαδική αναζήτηση χρειάζεται χρόνο $\mathcal{O}(m \log n)$ αφού κάθε σύγκριση μπορεί να πάρει χρόνο $\mathcal{O}(m)$ και έχουμε n προθήματα.

Χρησιμοποιώντας λίγο παραπάνω χώρο, ο χρόνο αυτός μπορεί να βελτιωθεί σε $\mathcal{O}(m + \log n)$.

Βιβλιογραφία

- Gusfield, Dan: ”Algorithms on Strings, Trees and Sequences: Computer Science and Computational Biology”. USA: Cambridge University Press, 1997.

- Richard Durbin, Sean R. Eddy, Anders Krogh, Graeme Mitchison: ”Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids”. Cambridge University Press, 1998.