Query-Driven Indexing for Scalable Peer-to-Peer Text Retrieval

Gleb Skobeltsyn, Toan Luu, Ivana Podnar Zarko, Martin Rajman, Karl Aberer

Περιγραφή του προβλήματος

- Ευρετηριοποίηση μεγάλων συλλογών εγγράφων πάνω σε ένα P2P δίκτυο
- Περιορισμοί:
  - Μικρή μνήμη στους κόμβους
  - Περιορισμένο bandwidth του δικτύου
- Σκοπός:
  - καλύτερη διαχείριση του ευρετηρίου
  - περιορισμός του όγκου πληροφορίας που διακινείται

Προσεγγίσεις

- Σε παλιότερες εργασίες διακρίνονται 2 προσεγγίσεις της λύσης του προβλήματος:
- HDK (*Highly Discriminative Keys*)
  - ευρετηριοποίηση όρων ή συνδυασμών όρων *(keys*) που εμφανίζονται συχνά με βάση ένα κατώφλι DFmax
- DCT (*Distributed Cache Table*)
  - caching των αποτελεσμάτων των πιο συχνών επερωτήσεων

Προσεγγίσεις (συνέχεια)

- Παρούσα προσέγγιση: συνδυασμός HDK και DCT.
- Στόχος:
  - μείωση του χώρου που καταλαμβάνεται με την HDK
  - περιορισμός των broadcast εκπομπών επερωτήσεων που είναι σπάνιες (κέρδος στο bandwidth).

Ιδέα

- Κάθε κόμβος:
  - διατηρεί ένα τμήμα Di της συνολικής συλλογής D και δημιουργεί ένα ευρετήριο πάνω σε αυτό
  - συμμετέχει στην συντήρηση και αποθήκευση του global ευρετηρίου το οποίο αφορά όλη τη συλλογή
- Global ευρετήριο: αποτελείται από ζεύγη της μορφής (k, PL(k)), όπου k το κλειδί και PL(k) η αντίστοιχη posting list
- Η κάθε posting list διατηρεί αναφορές για το πολύ DFmax έγγραφα.

Ιδέα (συνέχεια)

- Κάθε κόμβος Pi είναι υπεύθυνος για τη συντήρηση των κλειδιών που του έχουν ανατεθεί από τον κατανεμημένο πίνακα κατακερματισμού (DHT)
- Αρχικά, κάθε κόμβος κάνει την τοπική ευρετηριοποίηση και εισάγει τα κλειδιά που προκύπτουν στο δίκτυο
- Όταν λαμβάνει μια επερώτηση αρχίζει να την αναλύει σε υποσύνολα, ανάλογα με τους όρους που περιέχει (υποερωτήματα)

Ιδέα (συνέχεια)

- Για κάθε υποερώτημα q’ ο κόμβος ελέγχει αν το q’ σχετίζεται με κάποια posting list, ώστε να ανανεώσει την αναμενόμενη πιθανότητα χρήσης (EPU)
- Η EPU είναι μια μετρική του μοντέλου που δείχνει πόσο συχνό είναι ένα ερώτημα

Ιδέα (συνέχεια)

- Αν η EPU ξεπερνάει κάποιο κατώφλι EPUmin τότε το q’ θα πρέπει όπως λέμε να γίνει ενεργό κλειδί (κλειδί που αναζητείται συχνά)
- Οπότε δημιουργούμε και ευρετηριοποιούμε το νέο κλειδί και ενημερώνουμε τους κόμβους που έχουν έγγραφα που περιέχουν το νέο κλειδί
- Χρήση του μηχανισμού ONM (Opportunistic Notification Mechanism)

Μηχανισμός ONM

- Ιδέα: Shower multicast, δηλαδή η broadcast μετάδοση ‘σπάει’ σε multicast sessions τα οποία κάθε φόρα ‘ψαλιδίζουν’ τα έγγραφα με χαμηλή συχνότητα εμφάνισης των όρων
- Σκοπός: Αποτελεσματικότερη ενημέρωση των κόμβων-αποφυγή πλημμύρας
- Αναλυτική περιγραφή στο:

  A.Datta et al. Range Queries in Trie-Structured Overlays in P2P (2005)

Φιλτράρισμα

- Για τον περιορισμό του πλήθους και του μεγέθους των κλειδιών εφαρμόζουμε φιλτράρισμα με βάση:
  - Το μέγιστο μέγεθος ενός κλειδιού (s_{max})
  - Την ιδιότητα ένα κλειδί να είναι discriminative ή όχι
- Discriminative ονομάζεται ένα κλειδί k αν $df(k) \leq DF_{max}$ όπου DF_{max} μια παράμετρος του μοντέλου

Ιδιότητα DKs

- Ιδιότητα των discriminative κλειδιών(DKs):
  - Κάθε κλειδί που περιέχει ένα DK μικρότερου μεγέθους έχει και αυτό την ιδιότητα να είναι DK
- Πως μπορούμε να εκμεταλλευτούμε αυτή την ιδιότητα των DKs ώστε να μειώσουμε το μέγεθος του ευρετηρίου???

Αλγόριθμος ευρετηριοποίησης

- Ο αλγόριθμος τρέχει όταν ένα νέο κλειδί k γίνεται ενεργό
- Τότε κάθε κόμβος που έχει έγγραφα που περιέχουν όρους του k κάνει αναζήτηση στην τοπική συλλογή εγγράφων
- Αν το αποτέλεσμα δεν είναι κενό στέλνει τη λίστα των εγγράφων που έχουν ranking πάνω από minRank στον κόμβο που είναι υπεύθυνος για το k

Αλγόριθμος επεξεργασίας επερώτησης

- Σε κάθε κόμβο που τίθεται ένα ερώτημα q:
  - Αναλύεται το αρχικό q σε υποκλειδιά k
  - Για κάθε υποκλειδί k:
    - Αυξάνουμε το EPU(k)
    - Αν το k υπάρχει, ανακτούμε την posting list του
    - Αν όχι, μετράμε το EPU(k)
    - Αν ξεπερνά το κατώφλι EPUmin, τότε το δημιουργούμε και το προσθέτουμε στη λίστα νέων κλειδιών
    - Αν έχει συχνότητα εμφάνισης μικρότερη από DFmax, διαγράφουμε όλα τα νέα κλειδιά k' που προέκυψαν και για τα οποία ισχύει k≤k'
    - Τέλος, γίνεται ενημέρωση του DHT με χρήση του ONM

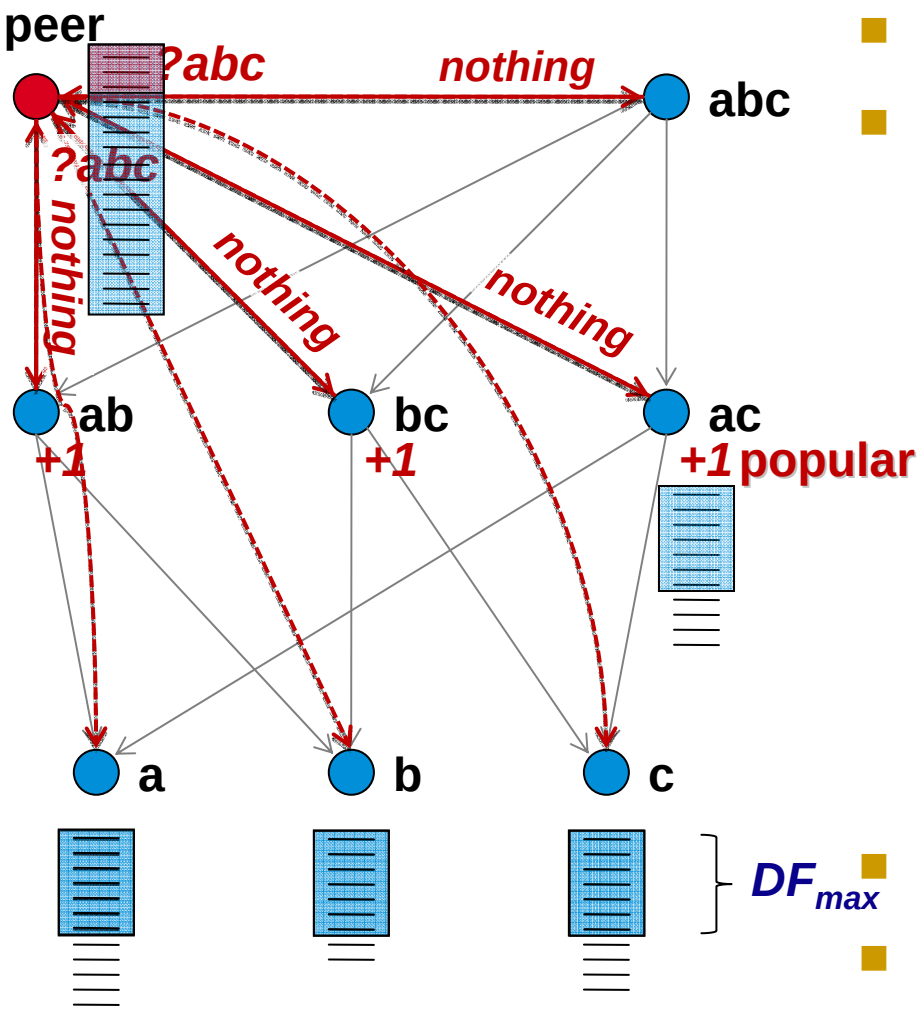
Παράδειγμα

- Το παράδειγμα που ακολουθεί είναι από παρουσίαση του Gleb Skobeltsyn

  Infoscale'07, June 6-8, 2007

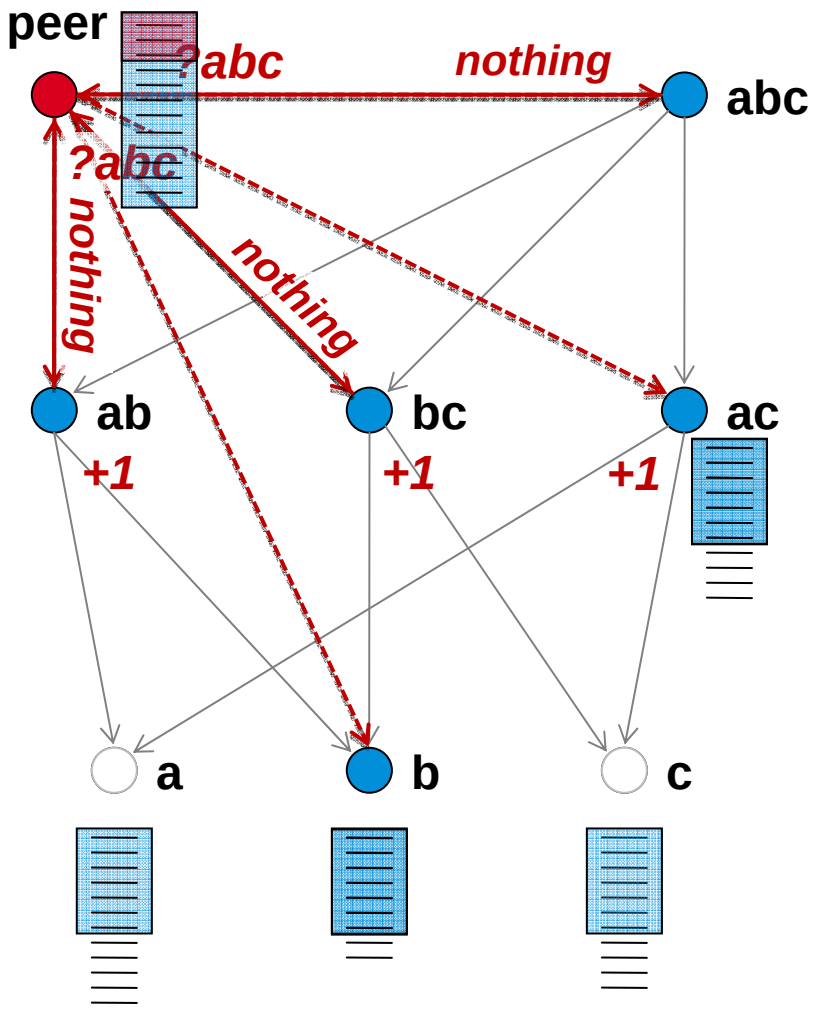
  Suzhou, China

Παράδειγμα



- Single term index is generated
- Process ***abc***
  1) Probe P_{abc}
  2) Probe P_{ab} P_{bc} and P_{ac}
  3) Probe P_a P_b and P_c
  4) Obtain top-DF_{max} results for ***a***, ***b*** and ***c*** (ranked w.r.t ***a***, ***b*** and ***c*** respectively)
  5) Contact peers in the list, re-rank the obtained results w.r.t ***abc***
  6) Output top-10
- Inc. the QF for ***ab***, ***bc*** and ***ac***
- Activate (index) ***ac***

Παράδειγμα (συνέχεια)



- Single term index is generated and ***ac*** is indexed
- Process ***abc***
  1) Probe P_{abc}
  2) Probe P_{ab} P_{bc} and P_{ac} – obtain the result for ***ac***
  3) Probe P_b and obtain the result for ***b***
  4) Contact all peers in the list to re-rank the obtained results w.r.t ***abc***
  5) Output top-10
- Inc. the QF for ***ab***, ***bc*** and ***ac***

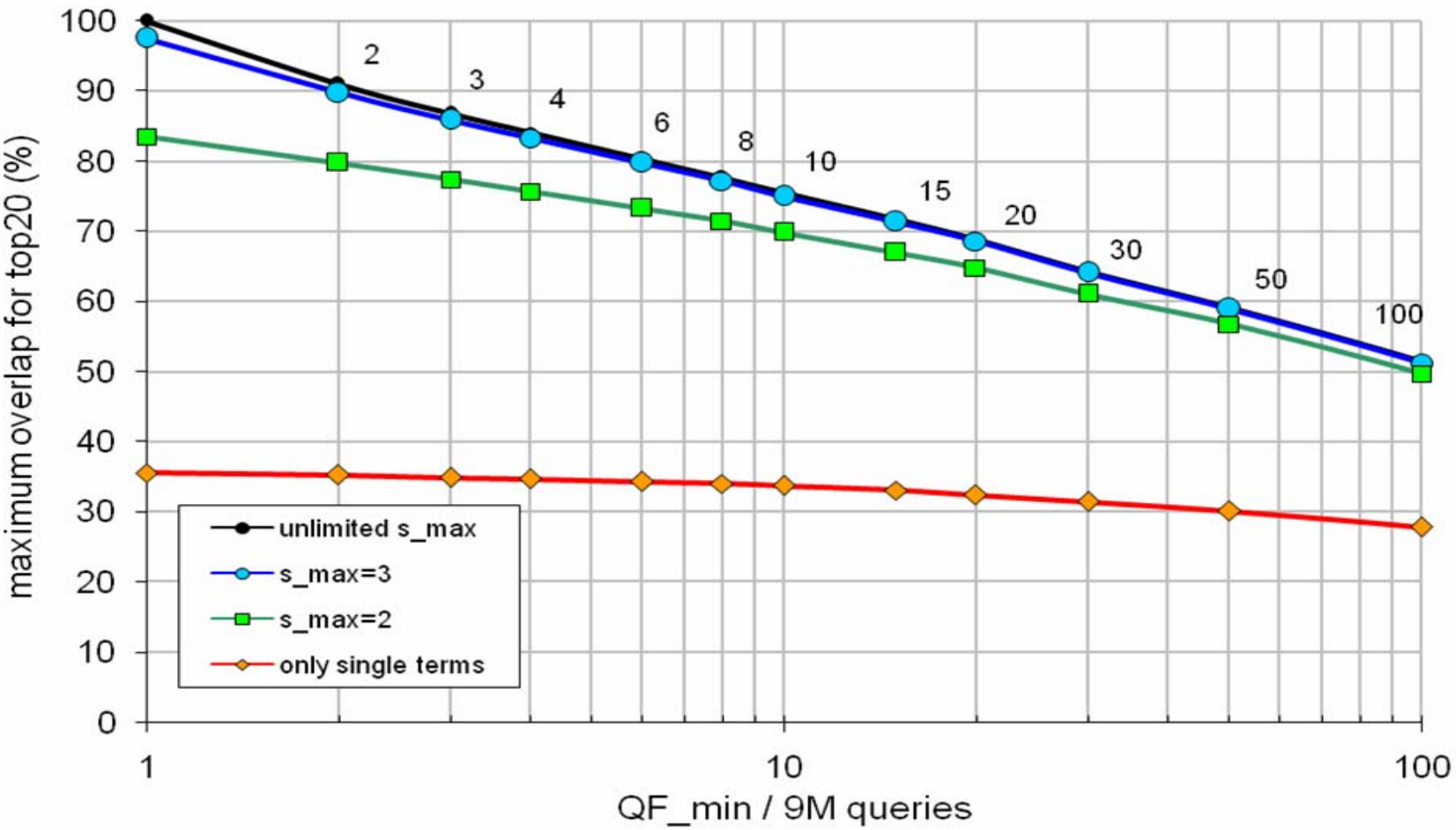
Scalability

- Με την παρούσα προσέγγιση το σύστημα είναι scalable αφού:
  - περιορίζουμε την τηλεπικοινωνιακή κίνηση με τη χρήση του ONM
  - κρατάμε χαμηλό το πλήθος των κλειδιών με χρήση του φιλτραρίσματος
  - οι posting lists που διακινούνται στο δίκτυο φράσσονται από την παράμετρο DF_{max}

Πείραμα 1

- Χρήση συνόλου επερωτήσεων που είχαν τεθεί στην Wikipedia
- Οι επερωτήσεις αυτές τέθηκαν στη μηχανή του Google και ανακτήθηκαν τα top-20 αποτελέσματα
- Κατόπιν, τα ίδια ερωτήματα τέθηκαν στο παρόν μοντέλο και έγινε έλεγχος επικάλυψης των αποτελεσμάτων με το Google
- Επικάλυψη σε ποσοστό έως και 80%
- Για μέγεθος κλειδιού >3 δεν παρατηρήθηκε ουσιαστική βελτίωση

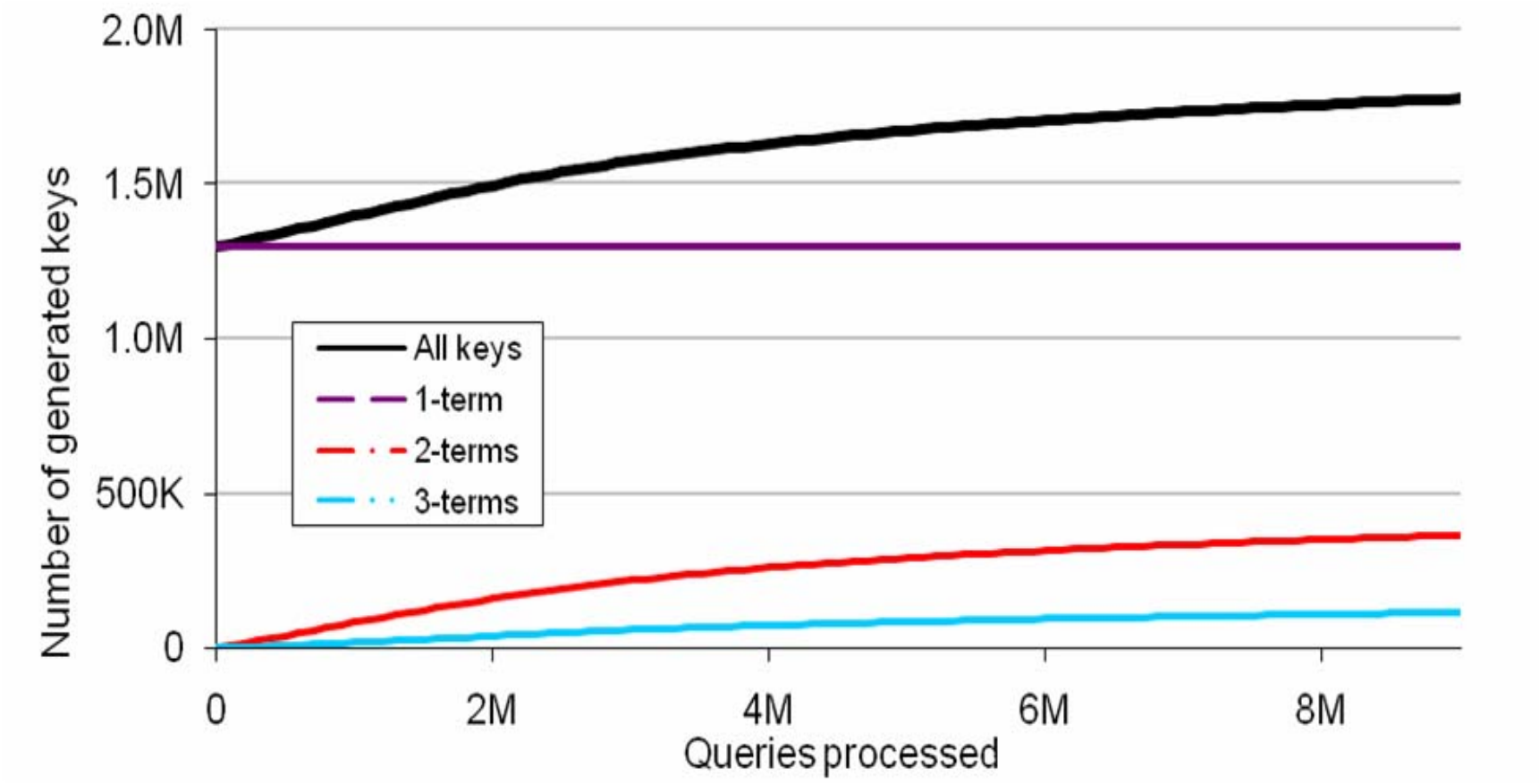
Πείραμα 1 (συνέχεια)



Πείραμα 2

- Μέτρηση του αριθμού των παραγόμενων κλειδιών κατά τη διάρκεια της επεξεργασίας των επερωτήσεων
- Παράμετροι DF_{max}= 100, EPU_{min}=4/(2*M) και s_{max}=3
- Μείωση του αριθμού των παραγόμενων κλειδιών σε σχέση με την HDK
- Συγκρίσεις επικάλυψης αντίστοιχες με το 1ο πείραμα, αλλά με τη μηχανή ανάκτησης Terrier
- Εξίσου ικανοποιητικά αποτελέσματα

Πείραμα 2 (συνέχεια)



Κριτική

- Πλεονεκτήματα
  - Μείωση του αριθμού των αποθηκευμένων κλειδιών
  - Μείωση της τηλεπικοινωνιακής κίνησης στο δίκτυο
  - Scalability
- Μειονεκτήματα
  - Δύσκολη υλοποίηση
  - Χρήση επερωτήσεων της Wikipedia για evaluation

Ερώτηση

- Πώς μπορούμε να εκμεταλλευτούμε την ιδιότητα των DKs: κάθε κλειδί που περιέχει ένα DK μικρότερου μεγέθους έχει και αυτό την ιδιότητα να είναι DK, ώστε να μειώσουμε το μέγεθος του ευρετηρίου???

Ευχαριστώ για την προσοχή σας

- Ερωτήσεις…???