

N. M. Μισυρλής

Τμήμα Πληροφορικής και Τηλεπικοινωνιών,
Πανεπιστήμιο Αθηνών

Βέλτιστα (στατικά) δυαδικά δένδρα αναζήτησης

Παράδειγμα: Σχεδιασμός προγράμματος μετάφρασης κειμένου από τα Ελληνικά στα Αγγλικά

Λύση:

- Για κάθε ελληνική λέξη που εμφανίζεται στο κείμενο, θα πρέπει να ανατρέχουμε στην αντίστοιχη αγγλική
- Κατασκευή δυαδικού δέντρου αναζήτησης (ΔΔΑ) το οποίο να χρησιμοποιεί ως κλειδιά n ελληνικές λέξεις και ως παρελκόμενα δεδομένα τις αντίστοιχες αγγλικές τους.

Στόχος: Ο ολικός χρόνος που δαπανάται στην αναζήτηση να είναι όσο το δυνατό μικρότερος

Βέλτιστα (στατικά) δυαδικά δένδρα αναζήτησης

Ερώτημα:

Πώς θα πρέπει να κατασκευάσουμε ένα δυαδικό δένδρο αναζήτησης ούτως ώστε να ελαχιστοποιήσουμε το πλήθος των κόμβων που διεξερχόμαστε σε όλες τις αναζητήσεις, αν γνωρίζουμε τη συχνότητα εμφάνισης της κάθε λέξης

Απάντηση:

Η δομή που χρειαζόμαστε λέγεται **Βέλτιστο (στατικό) δυαδικό δένδρο αναζήτησης**.

Βέλτιστα (στατικά) δυαδικά δένδρα αναζήτησης

• Το πρόβλημα:

- ▷ Δίνεται μια ταξινομημένη ακολουθία $K = \langle k_1, k_2, \dots, k_n \rangle$ από n διαφορετικά κλειδιά, έτσι ώστε $k_1 < k_2 < \dots < k_n$, με πιθανότητα αναζήτησης p_i για κάθε κλειδί k_i
- ▷ Δίνονται επίσης $n + 1$ "βουβά κλειδιά" d_0, d_1, \dots, d_n , τα οποία αντιπροσωπεύουν τιμές που δεν απαντούν στην ακολουθία K .
 - ★ Συγκεκριμένα, το κλειδί d_0 αντιπροσωπεύει όλες τις τιμές που είναι μικρότερες του k_1 , το d_n όλες τις τιμές που είναι μεγαλύτερες του k_n , και για $i = 1, 2, \dots, n - 1$, το βουβό κλειδί d_i αντιπροσωπεύει όλες τις τιμές που βρίσκονται μεταξύ του k_i και του k_{i+1} .
 - ★ Τα κλειδιά είναι εσωτερικοί κόμβοι και τα βουβά κλειδιά είναι καταληκτικοί κόμβοι (φύλλα)
 - ★ Για κάθε βουβό κλειδί d_i , έχουμε πιθανότητα αναζήτησης q_i
- ▷ Θέλουμε να κατασκευάσουμε ένα ΔΔΑ με το ελάχιστο αναμενόμενο κόστος αναζήτησης
- ▷ Για κάθε κλειδί k_i , κόστος $= \beta_T(k_i) + 1$, όπου $\beta_T(k_i)$ το βάθος ενός κόμβου στο δέντρο T .

Προσθέτουμε 1 στο κόστος γιατί το βάθος της ρίζας είναι 0

Αναμενόμενο κόστος αναζήτησης

- ▷ Το πραγματικό κόστος μιας αναζήτησης ισούται με το πλήθος των εξεταζόμενων κόμβων
- Οποιαδήποτε αναζήτηση είτε είναι επιτυχής (οπότε επιστρέφει κάποιο κλειδί k_i) είτε ανεπιτυχής (οπότε επιστρέφει κάποιο βουβό κλειδί d_i), έχει άθροισμα πιθανοτήτων ίσο με 1, δηλ.

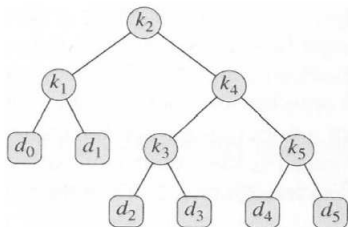
$$\sum_{i=1}^n p_i + \sum_{i=0}^n q_i = 1$$

- Το αναμενόμενο κόστος μιας αναζήτησης (κ.α) στο δέντρο T είναι

$$\begin{aligned} E[\text{κ.α. στο T}] &= \sum_{i=1}^n (\beta_T(k_i) + 1) \cdot p_i + \sum_{i=0}^n (\beta_T(d_i) + 1) \cdot q_i \\ &= 1 + \sum_{i=1}^n \beta_T(k_i) \cdot p_i + \sum_{i=0}^n \beta_T(d_i) \cdot q_i \end{aligned}$$

Παράδειγμα - Αναμενόμενο κόστος αναζήτησης

i	0	1	2	3	4	5
p_i		0.15	0.10	0.05	0.10	0.20
q_i	0.05	0.10	0.05	0.05	0.05	0.10



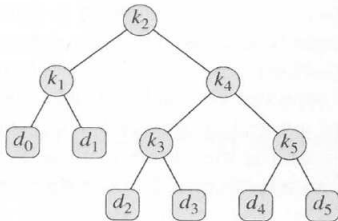
Κόστος: 2.80

Κόμβος	Βάθος	Πιθανότητα	Συνεισφορά
k_1	1	0.15	0.30
k_2	0	0.10	0.10
k_3	2	0.05	0.15
k_4	1	0.10	0.20
k_5	2	0.20	0.60
d_0	2	0.05	0.15
d_1	2	0.10	0.30
d_2	3	0.05	0.20
d_3	3	0.05	0.20
d_4	3	0.05	0.20
d_5	3	0.10	0.40
Σύνολο			2.80

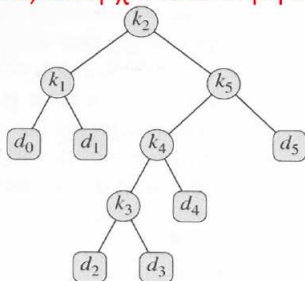
Παράδειγμα - Αναμενόμενο κόστος αναζήτησης

i	0	1	2	3	4	5
p_i		0.15	0.10	0.05	0.10	0.20
q_i	0.05	0.10	0.05	0.05	0.05	0.10

Αλλά, υπάρχει καλύτερη λύση:



Κόστος: 2.80



Κόστος: 2.75
Βέλτιστο!!

Παρατηρήσεις

- Το ζητούμενο είναι, για κάποιο δεδομένο σύνολο πιθανοτήτων να κατασκευάσουμε ένα ΔΔΑ το οποίο να έχει το ελάχιστο κόστος αναζήτησης.
- Ζητάμε, λοιπόν να κατασκευάσουμε ένα βέλτιστο ΔΔΑ
- Ένα βέλτιστο ΔΔΑ δεν είναι απαραίτητως δέντρο ελάχιστου ολικού ύψους
- Δεν είναι βέβαιο ότι μπορούμε να κατασκευάσουμε ένα βέλτιστο ΔΔΑ τοποθετώντας πάντοτε το κλειδί με τη μέγιστη πιθανότητα στη ρίζα.

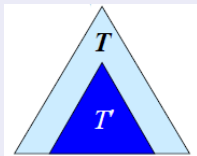
Εξαντλητικός έλεγχος

- Σήμανση των κόμβων οποιουδήποτε δυαδικού δέντρου n κόμβων με τα κλειδιά k_1, k_2, \dots, k_n ώστε να κατασκευάσουμε ένα ΔΔΑ
- Προσθήκη σε αυτό βουβών κλειδιών στη θέση καταληκτικών κόμβων

Ο εξαντλητικός έλεγχος όλων των δυνατοτήτων αποδίδει βραδύτατο αλγόριθμο. Το πλήθος των δυαδικών δέντρων με n κόμβους είναι $\Omega(4^n/n^{3/2})$, και επομένως για να εξαντλήσουμε όλες τις δυνατότητες θα πρέπει να εξετάσουμε εκθετικό πλήθος ΔΔΑ.

Βήμα 1: Η δομή ενός βέλτιστου ΔΔΑ

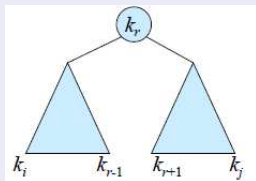
- Κάθε υποδέντρο του ΔΔΑ περιέχει διαδοχικά κλειδιά k_i, \dots, k_j , όπου $1 \leq i \leq j \leq n$.
- Επίσης θα περιέχει και τα βουβά κλειδιά d_{i-1}, \dots, d_j



- **Αν** το T είναι ένα βέλτιστο ΔΔΑ και **αν** το T περιλαμβάνει ένα υποδέντρο T' το οποίο περιέχει τα κλειδιά k_i, \dots, k_j και τα βουβά κλειδιά d_{i-1}, \dots, d_j **τότε** το υποδέντρο T' θα πρέπει να είναι επίσης βέλτιστο όσον αφορά το υποπρόβλημα με τα κλειδιά k_i, \dots, k_j και τα βουβά κλειδιά d_{i-1}, \dots, d_j
- **Απόδειξη:** Αποκοπή και επικόλληση.

Βέλτιστη υποδομή

- Κάποιο από τα κλειδιά k_i, \dots, k_j , έστω το k_r , όπου $i \leq r \leq j$, **θα αποτελεί τη ρίζα** ενός βέλτιστου υποδέντρου το οποίο θα πειέχει τα συγκεκριμένα κλειδιά
- Το αριστερό υποδέντρο της ρίζας k_r , θα περιέχει τα κλειδιά k_i, \dots, k_{r-1} (καθώς και τα βουβά κλειδιά d_{i-1}, \dots, d_{r-1})
- Το δεξί υποδέντρο της ρίζας k_r , θα περιέχει τα κλειδιά k_{r+1}, \dots, k_j (καθώς και τα βουβά κλειδιά d_r, \dots, d_j)



- **Για να βρούμε ένα βέλτιστο ΔΔΑ** πρέπει να
 - (i) εξετάσουμε όλες τις δυνατές ρίζες k_r , όπου $i \leq r \leq j$
 - (ii) προσδιορίσουμε όλα τα βέλτιστα ΔΔΑ που περιέχουν τα κλειδιά k_i, \dots, k_{r-1} και k_{r+1}, \dots, k_j

Βήμα 2: Μια αναδρομική λύση

- Εύρεση βέλτιστου ΔΔΑ για k_i, \dots, k_j , όπου $i \geq 1, j \leq n, j \geq i - 1$
- Τελικός μας στόχος είναι ο υπολογισμός του $e[1, n]$.
- Όταν $j = i - 1$, το δέντρο είναι κενό καθώς δεν υπάρχει κανένα πραγματικό κλειδί, αλλά μόνο το βουβό κλειδί d_{i-1} .
- Ορίζουμε $e[i, j] =$ αναμενόμενο κόστος αναζήτησης του βέλτιστου ΔΔΑ για k_i, \dots, k_j
- Αν $j = i - 1$, τότε $e[i, j] = a_{i-1}$
- Αν $j \geq i$,
 - ★ Επιλογή ρίζας k_r , για $i \leq r \leq j$
 - ★ Αναδρομική κατασκευή βέλτιστου ΔΔΑ
 - ▷ Για k_i, \dots, k_{r-1} , σαν αριστερό υποδέντρο είναι βέλτιστο ΔΔΑ
 - ▷ Για k_{r+1}, \dots, k_j , σαν δεξί υποδέντρο είναι βέλτιστο ΔΔΑ

Βήμα 2: Μια αναδρομική λύση

- Πώς μεταβάλλεται το αναμενόμενο κόστος αναζήτησης σε ένα υποδέντρο όταν αυτό καθίσταται υποδέντρο ενός κόμβου;
 - ★ Το βάθος του κάθε κόμβου στο υποδέντρο αυξάνεται κατά μια μονάδα
 - ★ Το αναμενόμενο κόστος αναζήτησης σε αυτό το υποδέντρο αυξάνεται κατά το άθροισμα όλων των πιθανοτήτων στο υποδέντρο:

$$w[i, j] = \sum_{l=i}^j p_l + \sum_{l=i-1}^j q_l$$

- Αν k_r είναι η ρίζα του βέλτιστου υποδέντρου το οποίο περιέχει τα κλειδιά k_i, \dots, k_j , τότε

$$\begin{aligned} e[i, j] &= p_r + (e[i, r-1] + w[i, r-1]) + (e[r+1, j] + w[r+1, j]) \\ &= e[i, r-1] + e[r+1, j] + w[i, j] \end{aligned}$$

γιατί

$$w[i, j] = w[i, r-1] + p_r + w[r+1, j].$$

Βήμα 2: Μια αναδρομική λύση

- Η επιλογή της ρίζας η οποία δίνει το ελάχιστο αναμενόμενο κόστος αναζήτησης δίνεται από την

$$e[i, j] = \begin{cases} q_{i-1}, & \text{αν } j = i - 1 \\ \min_{i \leq r \leq j} \{e[i, r-1] + e[r+1, j] + w[i, j]\}, & \text{αν } i \leq j \end{cases}$$

Οι τιμές των $e[i, j]$ αντιπροσωπεύουν τα αναμενόμενα κόστη αναζήτησης σε βέλτιστα ΔΔΑ.

- Για να διευκολύνουμε την πορεία κατασκευής ενός βέλτιστου ΔΔΑ, ορίζουμε ως *ρίζα* $[i, j]$, για $1 \leq i \leq j \leq n$, τον αύξοντα αριθμό r του κλειδιού k_r , το οποίο αποτελεί τη ρίζα ενός βέλτιστου ΔΔΑ που περιέχει τα κλειδιά k_i, \dots, k_j .

Βήμα 3: Υπολογίζοντας τη βέλτιστη λύση

Για κάθε υποπρόβλημα (i, j) , αποθηκεύουμε

- το αναμενόμενο κόστος αναζήτησης σε ένα πίνακα $e[1..n+1, 0..n]$
 - ▶ Θα χρησιμοποιήσουμε μόνο εισόδους $e[i, j]$, για τις οποίες $j \geq i - 1$
- $\text{ρίζα}[i, j]$ = ρίζα του υποδέντρου με κλειδιά k_i, \dots, k_j , για $1 \leq i \leq j \leq n$.
- $w[1..n+1, 0..n]$ = άθροισμα πιθανοτήτων:
 - ▶ $w[i, i-1] = q_{i-1}$, για $1 \leq i \leq n+1$ (βασική περίπτωση)
 - ▶ $w[i, j] = w[i, j-1] + p_j + q_j$, για $1 \leq i \leq j \leq n$.

Παρατήρηση: Αντί να υπολογίζουμε το $w[i, j]$ από το 0 κάθε φορά που υπολογίζουμε το $e[i, j]$, αποθηκεύουμε τις τιμές αυτές σε ένα πίνακα.

Βήμα 3: Υπολογισμός του αναμενόμενου κόστους αναζήτησης σε ένα βέλτιστο ΔΔΑ

OPTIMAL-BST(p, q, n)

1 let $e[1..n+1, 0..n]$, $w[1..n+1, 0..n]$,
and $root[1..n, 1..n]$ be new tables

2 for $i = 1$ to $n + 1$

3 $e[i, i - 1] = q_{i-1}$

4 $w[i, i - 1] = q_{i-1}$

5 for $l = 1$ to n

6 for $i = 1$ to $n - l + 1$

7 $j = i + l - 1$

8 $e[i, j] = \infty$

9 $w[i, j] = w[i, j - 1] + p_j + q_j$

10 for $r = i$ to j

11 $t = e[i, r - 1] + e[r + 1, j] + w[i, j]$

12 if $t < e[i, j]$

13 $e[i, j] = t$

14 $root[i, j] = r$

15 return e and $root$

Έστω όλα τα δέντρα με l κλειδιά

Καθορισμός του 1ου κλειδιού

Καθορισμός του 2ου κλειδιού

Προσδιορισμός της ρίζας
του βέλτιστου
(υπο)δέντρου.

Χρόνος = $O(n^3)$

Οι πίνακες $e[i, j]$, $w[i, j]$ και $\text{ρίζα}[i, j]$ που υπολογίζονται από τη διαδικασία ΒΕΛΤΙΣΤΟ ΔΔΑ για την κατανομή κλειδιών.

i	0	1	2	3	4	5
p_i		0.15	0.10	0.05	0.10	0.20
q_i	0.05	0.10	0.05	0.05	0.05	0.10

