

Λεξικό, Union – Find

Διδάσκοντες: **Σ. Ζάχος, Δ. Φωτάκης**

Επιμέλεια διαφανειών: **Δ. Φωτάκης**

Σχολή Ηλεκτρολόγων Μηχανικών
και Μηχανικών Υπολογιστών

Εθνικό Μετσόβιο Πολυτεχνείο



Πρόβλημα (ADT) Λεξικού

- Δυναμικά μεταβαλλόμενη συλλογή αντικειμένων που αναγνωρίζονται με κλειδί (π.χ. κατάλογοι, πίνακες ΒΔ).
- **Λεξικό** : συλλογή αντικειμένων με μοναδικό κλειδί.
 - Κλειδί: αριθμός ή τύπος δεδομένων με ολική διάταξη.
 - Γενίκευση και για μη-μοναδικά κλειδιά.
- ADT λεξικού υποστηρίζει ακολουθίες λειτουργιών:
 - Αναζήτηση στοιχείου με κλειδί x
 - $member(x)$: ελέγχει ύπαρξη στοιχείου με κλειδί x
 - $search(x)$: επιστρέφει δείκτη σε θέσεις x
 - Εισαγωγή στοιχείου με κλειδί x
 - Διαγραφή στοιχείου με κλειδί x

Λειτουργίες Λεξικού

- Λεξικό υποστηρίζει λειτουργίες:
 - Αναζήτηση/εισαγωγή/διαγραφή στοιχείου με κλειδί x
 - Εκτύπωση στοιχείων σε αύξουσα / φθίνουσα σειρά
 - Προηγούμενο και επόμενο στοιχείο.
 - Μέγιστο και ελάχιστο στοιχείο.
 - k -οστό μικρότερο στοιχείο
 - Βοηθητικές λειτουργίες ...

Υλοποιήσεις Λεξικού

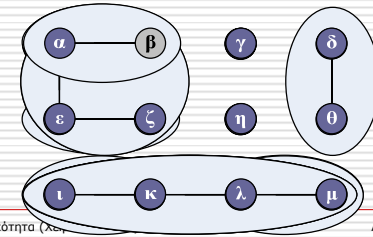
- Μη-ταξινομημένη διασυνδεδεμένη λίστα:
 - Εισαγωγή: $O(1)$
 - Αναζήτηση / διαγραφή: $O(n)$
 - Κατάλληλη όταν συχνές εισαγωγές, σπάνιες αναζητήσεις / μεμονωμένες διαγραφές (π.χ. log file).
- Ταξινομημένος πίνακας:
 - (Δυσδική) αναζήτηση: $O(\log n)$
 - Στατική συλλογή : «εισαγωγή» $O(\log n)$ / στοιχείο Χρόνος ταξινόμησης : $O(n \log n)$
 - Δυναμική συλλογή : εισαγωγή / διαγραφή $O(n)$
 - Κατάλληλη όταν συχνές αναζητήσεις και δεδομένα μεταβάλλονται σπάνια (π.χ. αγγλο-ελληνικό λεξικό).

Υλοποιήσεις Λεξικού

- Ζυγισμένο (Δυαδικό) Δέντρο Αναζήτησης:
 - Αναζήτηση / εισαγωγή / διαγραφή: $O(\log n)$
 - Μέγιστο / ελάχιστο / προηγούμενο / επόμενο / k -οστό: $O(\log n)$
 - Range queries σε γραμμικό χρόνο.
 - Πλήρως δυναμική – επιπλέον χώρος για δείκτες!
- Πίνακας Κατακερματισμού:
 - Αναζήτηση / διαγραφή: $O(1)$
 - Εισαγωγή: $O(1)$ expected amortized, $O(\log n)$ worst, $O(n)$ worst-case.
 - Δεν υποστηρίζει αποδοτικά άλλες λειτουργίες.
 - Δυναμική – επιπλέον χώρος στον πίνακα (util $\approx 50\%$)

Διαχείριση Διαμερίσεων Συνόλου

- Στοιχεία χωρίζονται σε ξένα υποσύνολα (κλάσεις ισοδυναμίας) που μεταβάλλονται δυναμικά με ένωση.
- Λειτουργίες:
 - Εύρεση $\text{find}(x)$: αντιπρόσωπο συνόλου όπου ανήκει x .
 - Ένωση $\text{union}(x, y)$: ένωση συνόλων όπου ανήκουν x και y .

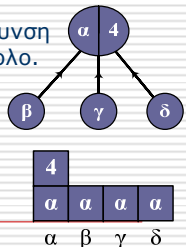


Πρόβλημα Union – Find

- Στοιχεία $U = \{1, 2, \dots, n\}$ αρχικά σε n ξένα σύνολα.
 - Κάθε σύνολο προσδιορίζεται από στοιχείο – αντιπρόσωπο.
- $\text{find}(x)$: αντιπρόσωπος συνόλου όπου ανήκει x .
 - Ελέγχουμε αν x και y ανήκουν στο ίδιο σύνολο.
 - Διατηρούμε μοναδικό αντιπρόσωπο για κάθε σύνολο.
- $\text{union}(x, y)$: αντικατάσταση (αντιπροσώπων) συνόλων x και y με σύνολο που προκύπτει από ένωση.
 - Νέος αντιπρόσωπος από τους αντιπροσώπους συνόλων x, y
 - Πάντα διαμέριση του U σε ξένα σύνολα.
 - $\leq n - 1$ ενώσεις (μετά από $n - 1$, ένα μόνο σύνολο).
- Δομή δεδομένων που ελαχιστοποιεί συνολικό χρόνο για ακολουθία m ερωτήσεων και $n - 1$ ενώσεων.

Αναπαράσταση με Δέντρα

- Σύνολο: δέντρο με ρίζα στοιχείο-αντιπρόσωπο.
 - size: μέγεθος συνόλου.
 - par ρίζας: στον εαυτό της (ή NULL ή 0).
- Στοιχείο: κόμβος με πεδία
 - id: όνομα στοιχείου.
 - par: δείκτης σε «επόμενο» (στην κατεύθυνση του αντιπροσώπου) στοιχείο στο ίδιο σύνολο.
- Υλοποίηση με πίνακα:
 - $A[x]$: par στοιχείου x .
 - Ρίζα – στοιχείο αντιπρόσωπος έχει $A[x] = x$ και πεδίο size.



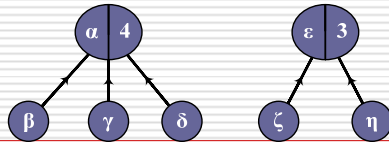
Αναπαράσταση με Δέντρα

□ **find(x)**: ακολουθούμε δείκτες **par** μέχρι ρίζα.

```
elem find(elem x) {
    while (x != A[x])
        x = A[x];
    return(x); }
```

□ **union(x, y)**: x και y αντιπρόσωποι **διαφορετικών συνόλων**

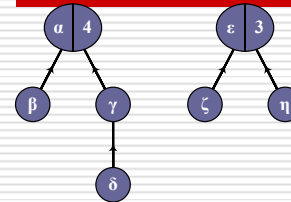
- **Συνένωση δέντρων**: **par** ρίζας 2^{ου} συν. = ρίζα 1^{ου} συν.
- **Ενημέρωση μεγέθους**



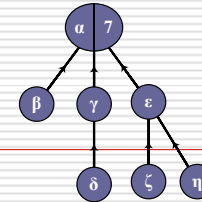
Αλγόριθμοι & Πολυπλοκότητα (Χειμώνας 2009)

Λεξικό, Union - Find 9

Ένωση



```
unionTree(elem x, elem y) {
    if (x == y) return;
    A[y] = x;
    A[x].size += A[y].size; }
```



Λεξικό, Union - Find 10

Απόδοση

- Χρόνος χ.π. για m finds και n unions: $O(mn + n)$
 - Union : $O(1)$ χρόνος.
 - Find : $O(\text{ύψος δέντρου})$
 - Χειρότερη περίπτωση: $\text{ύψος} = n - 1$
 - union(n-1, n), union(n-2, n-1), union(n-3, n-2), union(n-4, n-3), ..., union(3, 2), union(1, 2).
- Απλή δομή, εύκολη υλοποίηση, ακριβό find!
- Ισοδύναμη υλοποίηση με δέντρα και pointers.

Αλγόριθμοι & Πολυπλοκότητα (Χειμώνας 2009)

Λεξικό, Union - Find 11

Βεβαρυμένη Ένωση

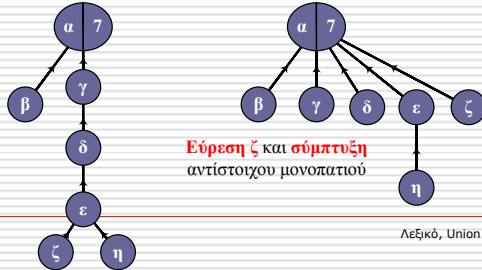
- Δεύτερο σύνολο με λιγότερα στοιχεία.
 - Λογαριθμικό ύψος δέντρου : $O(\log n)$.
 - Βεβαρυμένη ένωση: δέντρο ύψους h έχει $\geq 2^h$ στοιχεία.
- Απόδειξη με επαγωγή:
 - Ισχύει για $h = 0$ (δέντρο ενός στοιχείου).
 - Ένωση δέντρων x και y με ύψη h_x, h_y , και στοιχεία $s_x \geq s_y$
 - Επαγωγικά, υποθέτουμε $s \geq 2^h$ (για x και y)
 - Ύψος ένωσης = h_x : στοιχεία ένωσης $\geq 2^{\text{ύψος}}$
 - Ύψος ένωσης = $h_y + 1$: στοιχεία ένωσης $\geq 2s_y \geq 2^{\text{ύψος}}$
- Χρόνος χ.π. για m finds και n unions: $O(m \log n + n)$
 - Απλή υλοποίηση και αποδεκτή απόδοση.

Αλγόριθμοι & Πολυπλοκότητα (Χειμώνας 2009)

Λεξικό, Union - Find 12

Σύμπτυξη Μονοπατιών

- Find ακριβό γιατί στοιχεία μακριά από ρίζα.
- Σύμπτυξη μονοπατιού όταν $\text{find}(x)$:
 - Κάνουμε παιδιά ρίζας τους προγόνους του x και το x .
 - Δέντρο «κονταίνει» (ουσιαστικά χωρίς επιπλέον χρόνο).
 - Στο μέλλον, θα βρισκόμαστε σύνολο των στοιχείων γρήγορα.

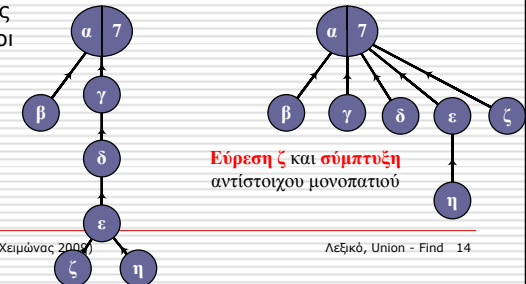


Λεξικό, Union - Find 13

Σύμπτυξη Μονοπατιών

```
elem findTreePathCompression(elem x) {  
    if (x != A[x])  
        A[x] = findTreePathCompression(A[x]);  
    return A[x];  
}
```

- Ανεβαίνουμε μέχρι ρίζα.
- Επιστρέφοντας μέχρι x , όλοι οι δείκτες par τίθενται να δείχνουν στη ρίζα.



Αλγόριθμοι & Πολυπλοκότητα (Χειμώνας 2009)

Λεξικό, Union - Find 14

Απόδοση

- Δέντρα, βεβαρυμένη ένωση, και σύμπτυξη μονοπατιών.
- Χρόνος χ.π. για $m \geq n$ finds και n unions: $O(m \alpha(n, m))$
 - $\alpha(n, m)$: αντίστροφη συνάρτηση Ackermann.
 - Μεγαλώνει εξαιρετικά αργά!
 - Στην πράξη, μπορεί να θεωρηθεί σταθερά.
- Απλή δομή, εύκολη υλοποίηση, και ουσιαστικά γραμμικός χρόνος!
- Εξάσκηση: $U = \{1, \dots, 16\}$.
 - Ενώσεις (2, 1), (3, 2), (4, 3), ..., (16, 15)
 - Δέντρα και με και χωρίς βεβαρυμένη ένωση.
 - Δέντρα: εύρεση 1 με και χωρίς σύμπτυξη μονοπατιών