



Δομές Δεδομένων

Ξένα Σύνολα

Α. Συμβώνης

Εθνικό Μετσόβιο Πολυτεχνείο

Σχολή Εφαρμοσμένων Μαθηματικών και Φυσικών Επιστημών

Τομέας Μαθηματικών



Ξένα Σύνολα

[Disjoint Sets]



Ξένα Σύνολα [Disjoint Sets]

- Ο ΑΤΔ `DisjointSet`

- ο Τα σύνολα αποτελούν διαμέριση του $U = \{0, 1, \dots, n-1\} \mid n \in \mathbb{N}$
 - ο Κάθε σύνολο προσδιορίζεται από ένα μέλος του, τον αντιπρόσωπό του (`representative`)
 - ο Ο αντιπρόσωπος του συνόλου ταυτίζεται με το όνομα του συνόλου
-
- ο `DisjointSet(int n):` Κατασκευαστής. Δημιουργεί n μονοσύνολα $\{0\}, \{1\}, \dots, \{n-1\}$
 - ο `find(int x):` Επιστρέφει το "σύνολο" στο οποίο ανήκει το στοιχείο x
 - ο `union(int x, int y):` Ένωση των "συνόλων" x και y



Ξένα Σύνολα [Disjoint Sets]

- Υλοποίηση του ΑΤΔ DisjointSet με διάνυσμα
 - Αποθηκεύεται στη θέση i ο αντιπρόσωπος του συνόλου στο οποίο ανήκει το i
 - $\text{find}(x) \rightarrow O(1)$
 - $\text{union}(x, y) \rightarrow O(n)$
 - Απλή υλοποίηση
 - Μια ακολουθία από $n - 1$ ενώσεις απαιτεί $O(n^2)$ χρόνο

Παράδειγμα: Απεικόνιση του συνόλου D για $n = 8$

$$D = \{ \{0, 1, \mathbf{2}, 4, 5\}, \\ \{\mathbf{3}, 7\}, \\ \{\mathbf{6}\} \\ \}$$

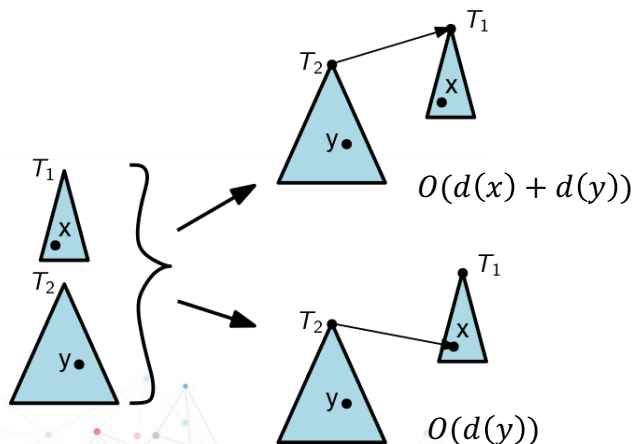
0	1	2	3	4	5	6	7
2	2	2	3	2	2	6	3



Ξένα Σύνολα [Disjoint Sets]

Υλοποίηση του ΑΤΔ DisjointSet με δένδρα

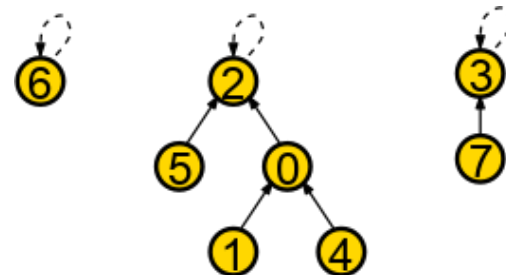
- Τα σύνολα είναι οργανωμένα σε δένδρα όπου η ρίζα είναι ο αντιπρόσωπος του συνόλου
 - $\text{find}(x) \rightarrow O(d(x))$, όπου το $d(x)$ είναι το βάθος του x στο δένδρο στο οποίο ανήκει
 - $\text{union}(x, y) \rightarrow O(d(x) + d(y))$



Παράδειγμα

$D = \{ \{0, 1, \mathbf{2}, 4, 5\}, \{ \mathbf{3}, 7\}, \{ \mathbf{6} \} \}$

- Αναπαράσταση με δείκτες:



- Αναπαράσταση με διάνυσμα:

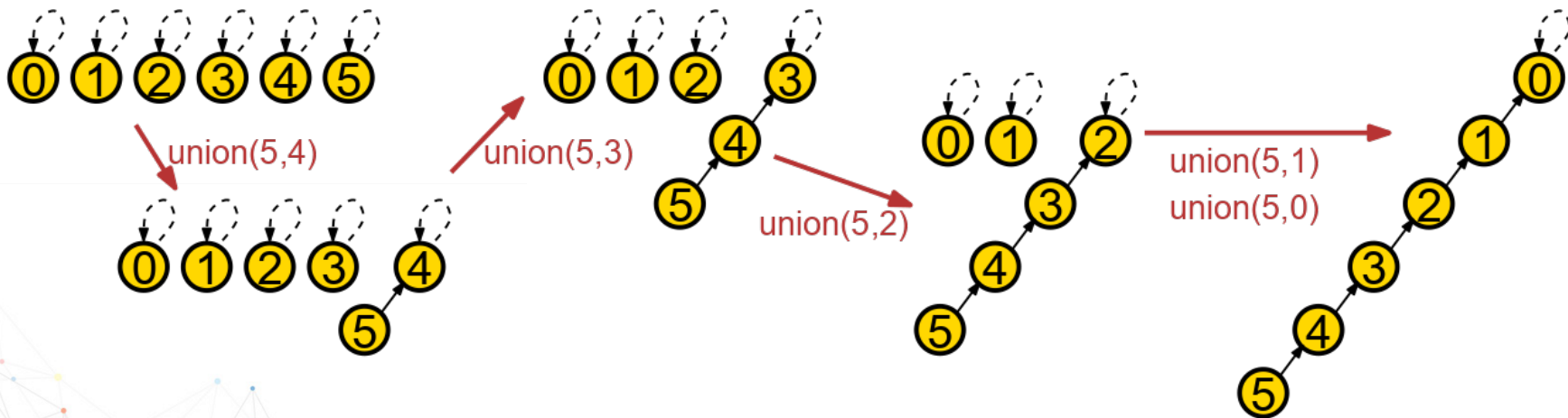
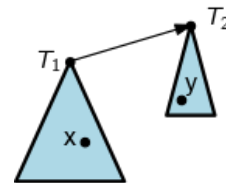
0	1	2	3	4	5	6	7
2	0	2	3	0	2	6	3



Ξένα Σύνολα [Disjoint Sets]

• Ένωση συνόλων (1^η προσέγγιση)

- Ο αντιπρόσωπος της ένωσης γίνεται ο αντιπρόσωπος του μικρότερου συνόλου
- Το ύψος του δένδρου μπορεί να είναι $O(n)$
- Οι $n - 1$ διαδοχικές ενώσεις απαιτούν $O(n^2)$ χρόνο

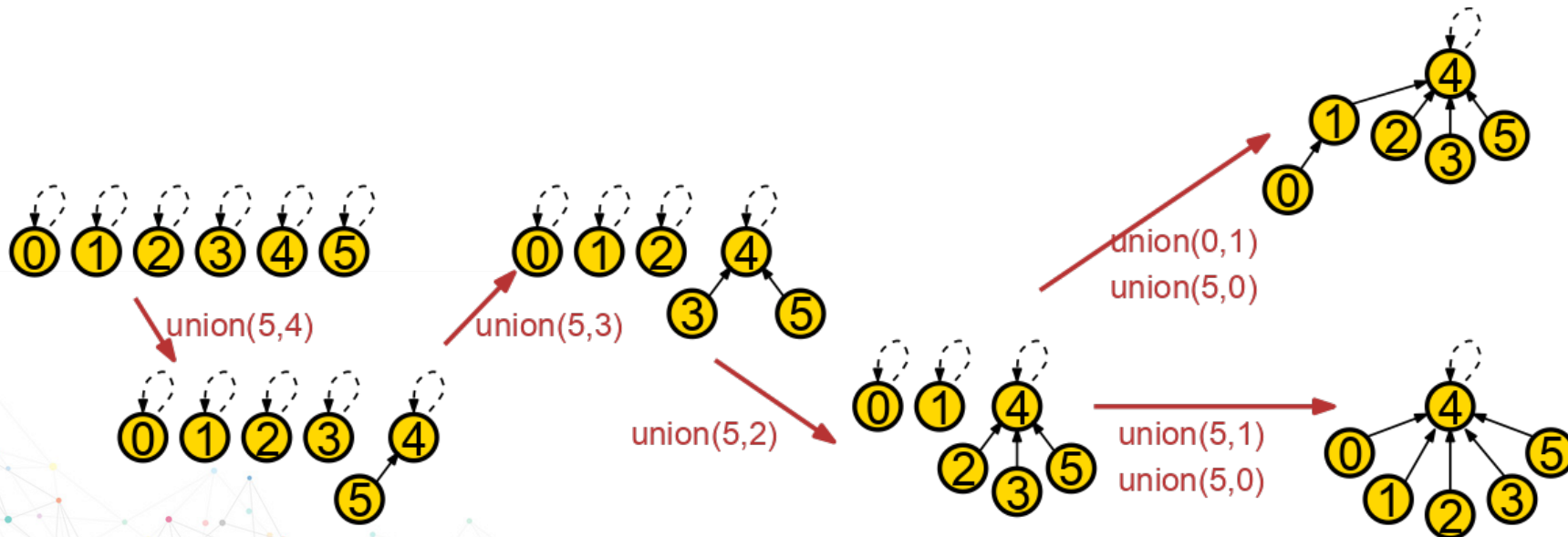
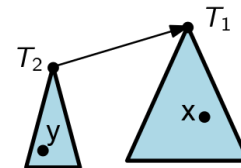




Ξένα Σύνολα [Disjoint Sets]

- Ένωση συνόλων (2^η προσέγγιση)

- Ο αντιπρόσωπος της ένωσης γίνεται ο αντιπρόσωπος του μεγαλύτερου συνόλου
- Ερώτηση: Ποιο είναι το μέγιστο ύψος ενός δένδρου;

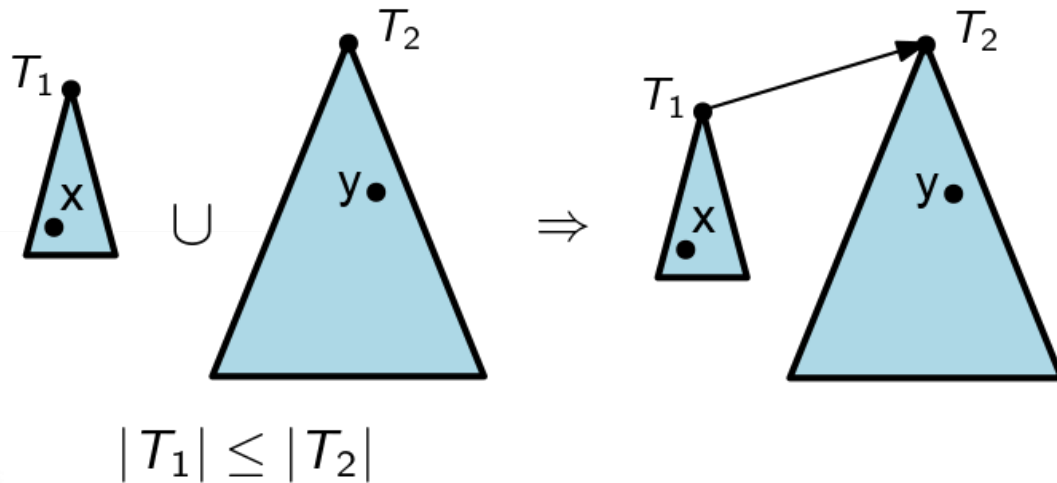




Ξένα Σύνολα [Disjoint Sets]

- Κανόνας Βεβαρημένης Ένωσης [Weighted Union Rule]

- Βάρος ενός συνόλου είναι το πλήθος των στοιχείων του
- Ο αντιπρόσωπος του ελαφρύτερου συνόλου συσχετίζεται με αυτόν του βαρύτερου συνόλου





Ξένα Σύνολα [Disjoint Sets]

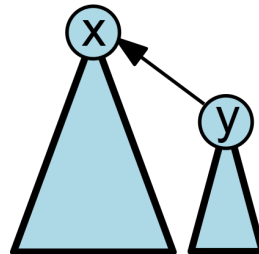
Λήμμα: Για κάθε στοιχείο x , το πλήθος των απογόνων του $w(x)$ είναι τουλάχιστον $2^{h(x)}$, όπου $h(x)$ το ύψος του στο δάσος των δένδρων που έχει δημιουργηθεί μέχρι στιγμής.

Απόδειξη: (Με επαγωγή στο ύψος των δένδρων)

- Βάση: $h(x) = 0 \Rightarrow w(x) \geq 2^0 = 1$, άρα ισχύει
- Επαγωγικό υπόθεση: Έστω ότι για κάθε κόμβο x με ύψος $h(x) < k$ ισχύει: $w(x) \geq 2^{h(x)}$
- Επαγωγικό βήμα: Θα δείξουμε ότι η ιδιότητα ισχύει για κάθε κόμβο x με $h(x) = k$
Έστω x κόμβος με $h(x) = k > 0$ και y το παιδί που ενώθηκε με τον x , όπου $h(y) \leq h(x)$
 - $w'(x)$: το βάρος του x **πριν** την ένωση
 - $w(y) = w'(y)$
 - $w'(x) \geq w'(y)$ [weighted union]

$$\begin{aligned}\text{Άρα: } 2^{h(x)} &= 2^{\max\{1+h(y), h'(x)\}} \\ &= \max\{2^{h(y)+1}, 2^{h'(x)}\} \\ &\leq \max\{2w(y), w'(x)\} \\ &\leq \max\{w(x), w(x)\} = w(x)\end{aligned}$$

$$\Rightarrow 2^{h(x)} \leq w(x)$$





Ξένα Σύνολα [Disjoint Sets]

Θεώρημα: Το ύψος του δένδρου ενός συνόλου n στοιχείων είναι το πολύ $\log n$

Απόδειξη:

- Έστω x το στοιχείο στη ρίζα του δένδρου
 - $w(x) \geq 2^{h(x)}$ (Λήμμα)
 - $n = w(x)$ (Υπόθεση)
- $$\left. \begin{array}{l} \bullet w(x) \geq 2^{h(x)} \text{ (Λήμμα)} \\ \bullet n = w(x) \text{ (Υπόθεση)} \end{array} \right\} \Rightarrow n \geq 2^{h(x)} \Rightarrow h(x) \leq \log_2 n$$

Θεώρημα: Μια αναμεμειγμένη ακολουθία από $n - 1$ ενώσεις και m αναζητήσεις κοστίζει $O(n \log n + m \log n)$ χρόνο, εφ' όσον χρησιμοποιηθεί ο κανόνας βεβαρημένης ένωσης.

Απόδειξη:

- $n - 1$ ενώσεις: $\rightarrow O(n \log n)$
 - m αναζητήσεις: $\rightarrow O(m \log n)$
- $$\left. \begin{array}{l} \bullet n - 1 \text{ ενώσεις: } \rightarrow O(n \log n) \\ \bullet m \text{ αναζητήσεις: } \rightarrow O(m \log n) \end{array} \right\} \Rightarrow O(\mathbf{n \log n} + m \log n)$$

- Μετά την εκτέλεση μιας ένωσης γίνεται ενημέρωση του “μεγέθους” του νέου συνόλου
- Το μέγεθος κάθε συνόλου αποθηκεύεται στη ρίζα του αντίστοιχου δένδρου
- Ο προσδιορισμός του μεγέθους ενός συνόλου απαιτεί κλήση της μεθόδου `find()`