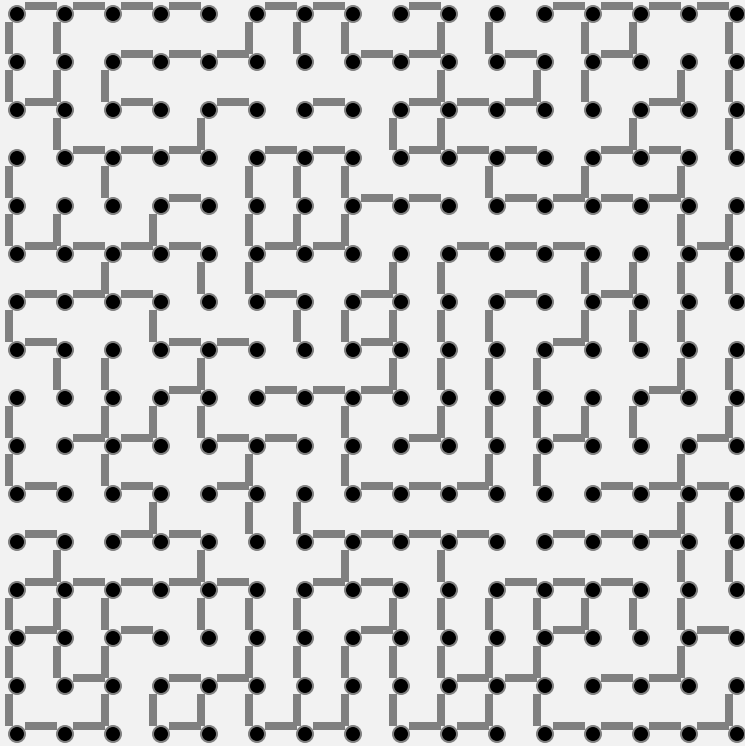
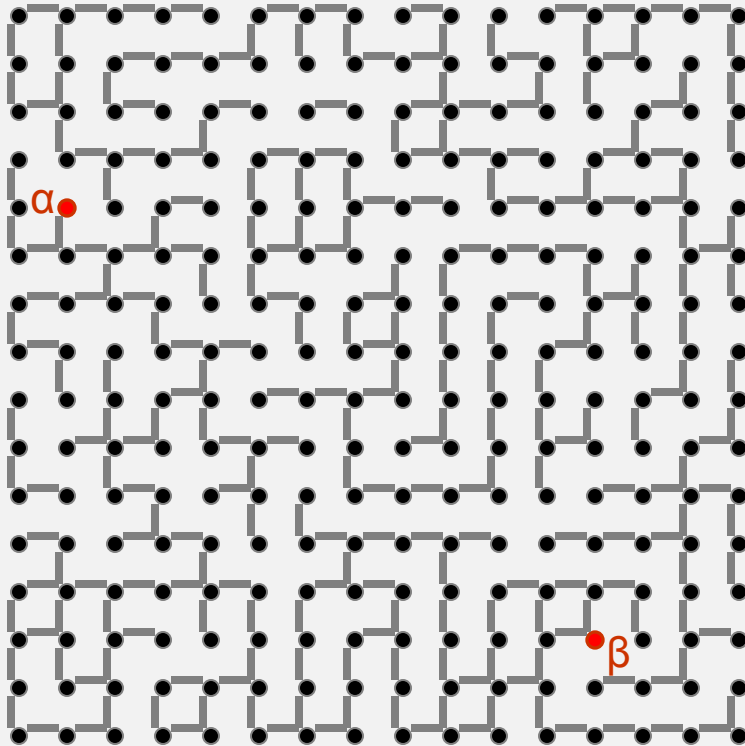


Συνδετικότητα γραφήματος (graph connectivity)



Συνδετικότητα γραφήματος (graph connectivity)

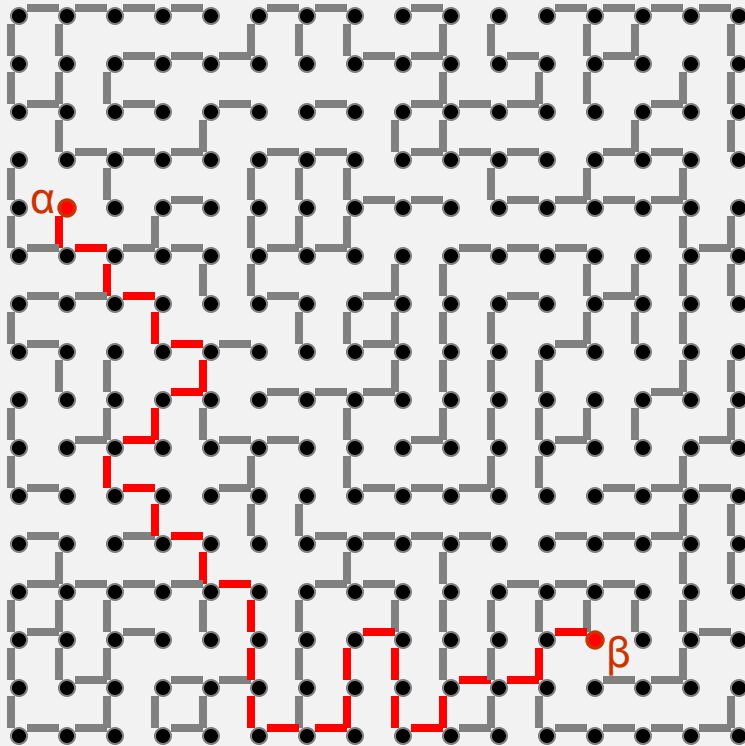


Υπάρχει μονοπάτι μεταξύ α και β;

Παραδείγματα:

- υπολογιστές ενός δικτύου
- ιστοσελίδες
- ισοδύναμες μεταβλητές ενός προγράμματος
- τρανζίστορ σε ηλεκτρονικό κύκλωμα

Συνδετικότητα γραφήματος (graph connectivity)

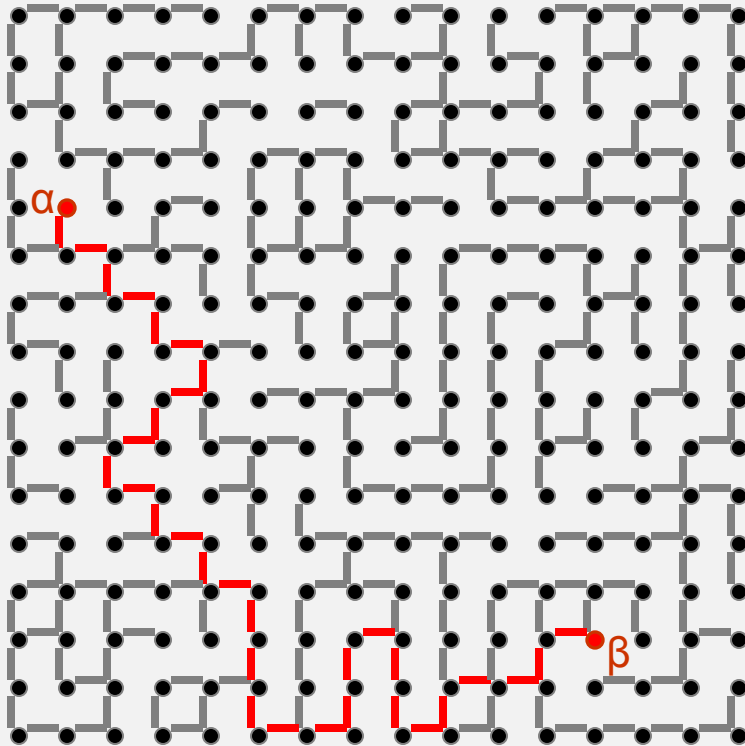


Υπάρχει μονοπάτι μεταξύ α και β;

Παραδείγματα:

- υπολογιστές ενός δικτύου
- ιστοσελίδες
- ισοδύναμες μεταβλητές ενός προγράμματος
- τρανζίστορ σε ηλεκτρονικό κύκλωμα

Συνδετικότητα γραφήματος (graph connectivity)



Υπάρχει μονοπάτι μεταξύ α και β ;

Παραδείγματα:

- υπολογιστές ενός δικτύου
- ιστοσελίδες
- ισοδύναμες μεταβλητές ενός προγράμματος
- τρανζίστορ σε ηλεκτρονικό κύκλωμα

Αν το γράφημα είναι γνωστό εξ αρχής (offline) τότε μπορούμε να λύσουμε το πρόβλημα σε γραμμικό χρόνο με οριζόντια ή καθοδική διερεύνηση. Θα δώσουμε μια εναλλακτική λύση η οποία μπορεί να χρησιμοποιηθεί ακόμα και όταν το γράφημα μας δίνεται μια ακμή τη φορά (online).

Σχέση ισοδυναμίας (equivalence relation)

Η συδετικότητα σε ένα γράφημα ορίζει μία σχέση ισοδυναμίας μεταξύ των κόμβων του γραφήματος

Σ = Διμερής σχέση μεταξύ αντικειμένων ενός συνόλου A ; υποσύνολο του $A \times A$

Σ είναι σχέση ισοδυναμίας (equivalence relation):

Αυτοπαθής: $(\alpha, \alpha) \in \Sigma, \forall \alpha \in A$

Συμμετρική: $(\alpha, \beta) \in \Sigma \Rightarrow (\beta, \alpha) \in \Sigma$

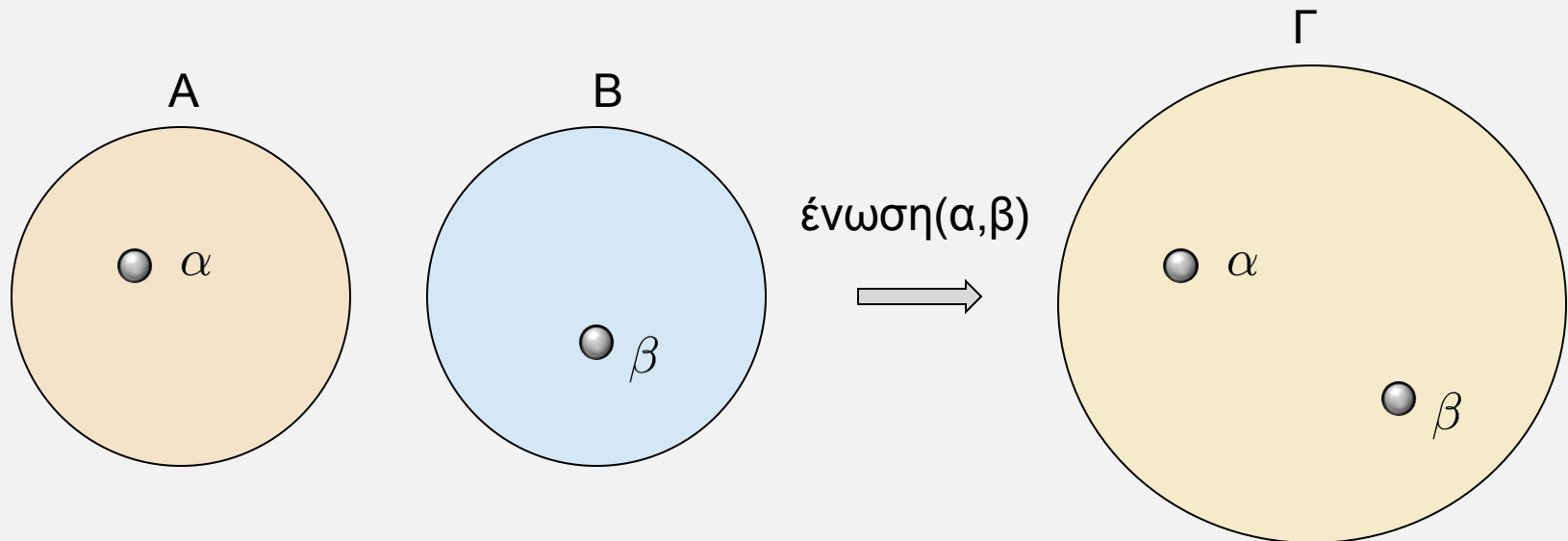
Μεταβατική: $(\alpha, \beta) \in \Sigma$ και $(\beta, \gamma) \in \Sigma \Rightarrow (\alpha, \gamma) \in \Sigma$

Το πρόβλημα ένωσης-εύρεσης (union-find)

Θέλουμε μία δομή δεδομένων που να υποστηρίζει τις παρακάτω λειτουργίες:

ένωση(α, β): αντικαθιστά τα σύνολα που περιέχουν τα α και β με την ένωση τους

εύρεση(α): επιστρέφει το όνομα του συνόλου που περιέχει το α



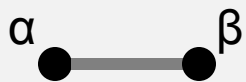
Το πρόβλημα ένωσης-εύρεσης (union-find)

Θέλουμε μία δομή δεδομένων που να υποστηρίζει τις παρακάτω λειτουργίες:

ένωση(α, β): αντικαθιστά τα σύνολα που περιέχουν τα α και β με την ένωση τους

εύρεση(α): επιστρέφει το όνομα του συνόλου που περιέχει το α

Η δομή αυτή αρκεί για να επιλύσουμε το πρόβλημα τις συνδετικότητας:



Για κάθε ακμή (α, β) εκτελούμε $\text{ένωση}(\alpha, \beta)$



Το α συνδέεται με το β αν και μόνο αν $\text{εύρεση}(\alpha) = \text{εύρεση}(\beta)$

Δομή Εύρεσης-Ένωσης με Λίστες

Διατηρούμε ξένα μεταξύ τους σύνολα $S_1, S_2, \dots, S_\ell$ που είναι υποσύνολα ενός συνόλου $K = \{k_1, k_2, \dots, k_n\}$, όπου $S_i \cap S_j = \emptyset, i \neq j$ και $\bigcup_{i=1}^\ell S_i = K$

Λειτουργίες

$make_set(k) =$ Δημιουργεί νέο σύνολο με μόνο στοιχείο το k

$unite(x, y) =$ Έστω S το σύνολο που περιέχει το x και έστω T το σύνολο που περιέχει το y . Δημιουργεί ένα νέο σύνολο $R = S \cup T$ και διαγράφει τα S και T

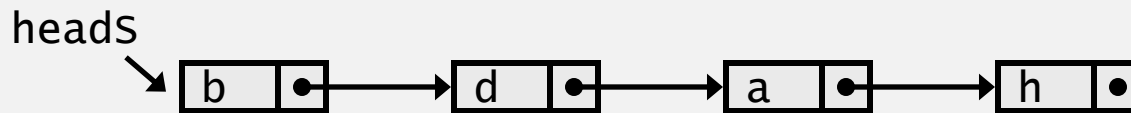
$find(x) =$ Επιστρέφει το όνομα του συνόλου που περιέχει το x

Ως «όνομα» ενός συνόλου S μπορούμε να ορίσουμε ένα στοιχείο που να αποτελεί τον «αντιπρόσωπο» του συνόλου

Δομή Εύρεσης-Ένωσης με Λίστες

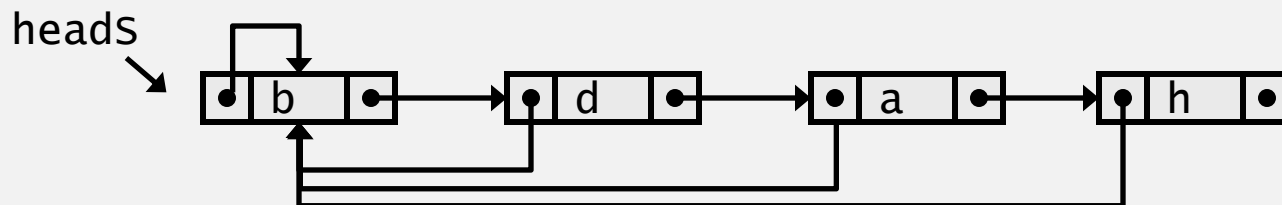
Κάθε σύνολο S μπορεί να αποθηκευτεί σε μια συνδεδεμένη λίστα με τον αντιπρόσωπο του στην κορυφή της λίστας

Π.χ. για το σύνολο $S = \{a, b, d, h\}$ με αντιπρόσωπο b



Για να έχουμε τη δυνατότητα να βρούμε τον αντιπρόσωπο γρήγορα, προσθέτουμε στη δομή Node ένα ακόμα πεδίο `rep` που είναι αναφορά στον κόμβο-αντιπρόσωπο

```
private class Node {Item item; Node next; Node rep};
```



Δομή Εύρεσης-Ένωσης με Λίστες

Κάθε σύνολο S μπορεί να αποθηκευτεί σε μια συνδεδεμένη λίστα με τον αντιπρόσωπο του στην κορυφή της λίστας

Για να έχουμε τη δυνατότητα να βρούμε τον αντιπρόσωπο γρήγορα, προσθέτουμε στη δομή `Node` ένα ακόμα πεδίο `rep` που είναι αναφορά στον κόμβο-αντιπρόσωπο

Τώρα τα ορίσματα που δέχονται και οι τιμές που επιστρέφουν οι λειτουργίες $find(x)$ και $unite(x, y)$ είναι αναφορές σε κόμβους

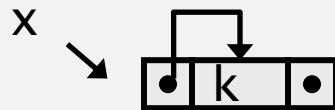
Η λειτουργία $makeset(k)$ επιστρέφει αναφορά x σε νέο κόμβο που αποθηκεύει το στοιχείο k

Δομή Εύρεσης-Ένωσης με Λίστες

Κάθε σύνολο S μπορεί να αποθηκευτεί σε μια συνδεδεμένη λίστα με τον αντιπρόσωπο του στην κορυφή της λίστας

Για να έχουμε τη δυνατότητα να βρούμε τον αντιπρόσωπο γρήγορα, προσθέτουμε στη δομή `Node` ένα ακόμα πεδίο `rep` που είναι αναφορά στον κόμβο-αντιπρόσωπο

$makeset(k)$: Δημιουργεί νέο κόμβο που αποθηκεύει το στοιχείο k και επιστρέφει αναφορά x στον κόμβο



```
Node x = new Node();  
x.item = k;  
x.next = null;  
x.rep = x;
```

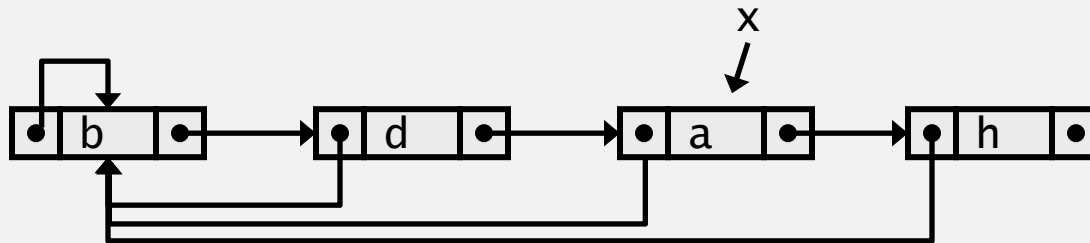
Χρόνος $O(1)$

Δομή Εύρεσης-Ένωσης με Λίστες

Κάθε σύνολο S μπορεί να αποθηκευτεί σε μια συνδεδεμένη λίστα με τον αντιπρόσωπο του στην κορυφή της λίστας

Για να έχουμε τη δυνατότητα να βρούμε τον αντιπρόσωπο γρήγορα, προσθέτουμε στη δομή `Node` ένα ακόμα πεδίο `rep` που είναι αναφορά στον κόμβο-αντιπρόσωπο

$find(x)$: Επιστρέφει την αναφορά `x.rep`



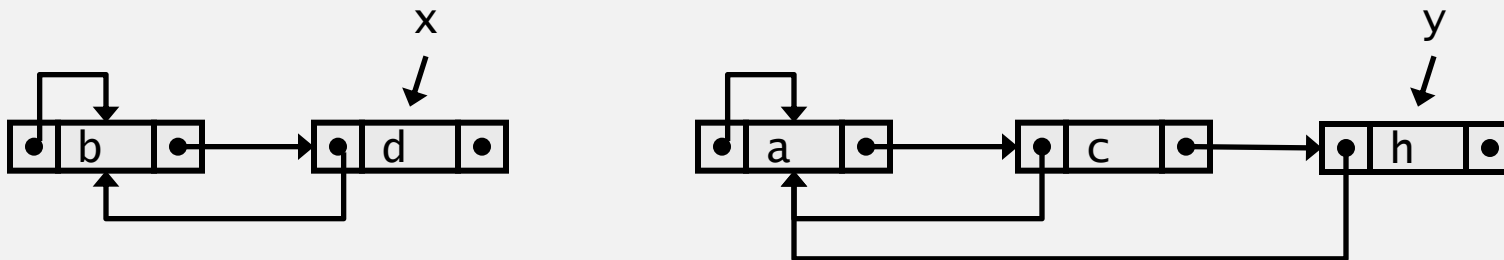
Χρόνος $O(1)$

Δομή Εύρεσης-Ένωσης με Λίστες

Κάθε σύνολο S μπορεί να αποθηκευτεί σε μια συνδεδεμένη λίστα με τον αντιπρόσωπο του στην κορυφή της λίστας

Για να έχουμε τη δυνατότητα να βρούμε τον αντιπρόσωπο γρήγορα, προσθέτουμε στη δομή `Node` ένα ακόμα πεδίο `rep` που είναι αναφορά στον κόμβο-αντιπρόσωπο

$unite(x, y)$: Έστω $p = x.rep$ και $q = y.rep$. Διατρέχει τη λίστα του q και για κάθε κόμβο v αυτής της λίστας θέτει $v.rep = p$. Τέλος συνενώνει τις δύο λίστες σε μία λίστα με αρχικό κόμβο το p .



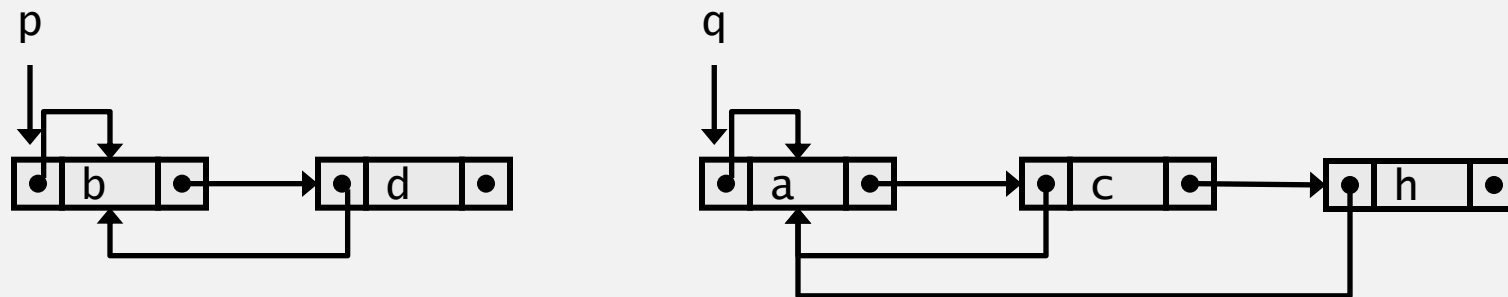
Χρόνος $O(n_y)$ όπου n_y ο αριθμός των στοιχείων στο σύνολο που περιέχει το y

Δομή Εύρεσης-Ένωσης με Λίστες

Κάθε σύνολο S μπορεί να αποθηκευτεί σε μια συνδεδεμένη λίστα με τον αντιπρόσωπο του στην κορυφή της λίστας

Για να έχουμε τη δυνατότητα να βρούμε τον αντιπρόσωπο γρήγορα, προσθέτουμε στη δομή `Node` ένα ακόμα πεδίο `rep` που είναι αναφορά στον κόμβο-αντιπρόσωπο

$unite(x, y)$: Έστω $p=x.rep$ και $q=y.rep$. Διατρέχει τη λίστα του q και για κάθε κόμβο v αυτής της λίστας θέτει $v.rep = p$. Τέλος συνενώνει τις δύο λίστες σε μία λίστα με αρχικό κόμβο το p .



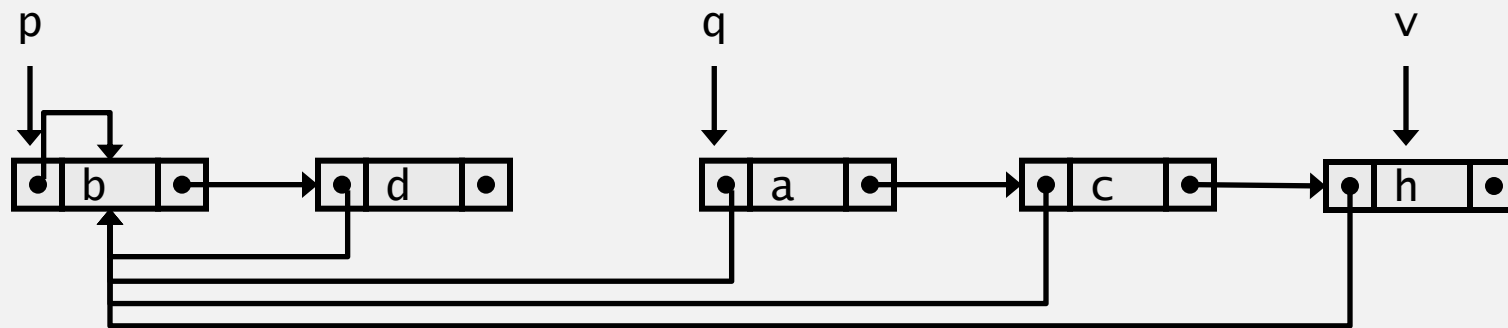
Χρόνος $O(n_y)$ όπου n_y ο αριθμός των στοιχείων στο σύνολο που περιέχει το y

Δομή Εύρεσης-Ένωσης με Λίστες

Κάθε σύνολο S μπορεί να αποθηκευτεί σε μια συνδεδεμένη λίστα με τον αντιπρόσωπο του στην κορυφή της λίστας

Για να έχουμε τη δυνατότητα να βρούμε τον αντιπρόσωπο γρήγορα, προσθέτουμε στη δομή `Node` ένα ακόμα πεδίο `rep` που είναι αναφορά στον κόμβο-αντιπρόσωπο

$unite(x, y)$: Έστω $p=x.rep$ και $q=y.rep$. Διατρέχει τη λίστα του q και για κάθε κόμβο v αυτής της λίστας θέτει $v.rep = p$. Τέλος συνενώνει τις δύο λίστες σε μία λίστα με αρχικό κόμβο το p .



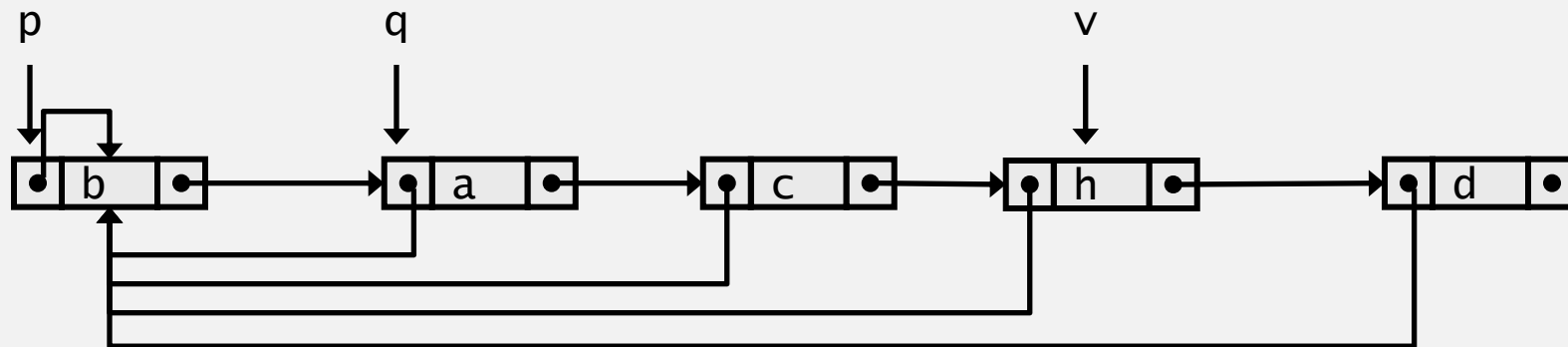
Χρόνος $O(n_y)$ όπου n_y ο αριθμός των στοιχείων στο σύνολο που περιέχει το y

Δομή Εύρεσης-Ένωσης με Λίστες

Κάθε σύνολο S μπορεί να αποθηκευτεί σε μια συνδεδεμένη λίστα με τον αντιπρόσωπο του στην κορυφή της λίστας

Για να έχουμε τη δυνατότητα να βρούμε τον αντιπρόσωπο γρήγορα, προσθέτουμε στη δομή `Node` ένα ακόμα πεδίο `rep` που είναι αναφορά στον κόμβο-αντιπρόσωπο

$unite(x, y)$: Έστω $p = x.rep$ και $q = y.rep$. Διατρέχει τη λίστα του q και για κάθε κόμβο v αυτής της λίστας θέτει $v.rep = p$. Τέλος συνενώνει τις δύο λίστες σε μία λίστα με αρχικό κόμβο το p .



Χρόνος $O(n_y)$ όπου n_y ο αριθμός των στοιχείων στο σύνολο που περιέχει το y

Δομή Εύρεσης-Ένωσης με Λίστες

Κάθε σύνολο S μπορεί να αποθηκευτεί σε μια συνδεδεμένη λίστα με τον αντιπρόσωπο του στην κορυφή της λίστας

Για να έχουμε τη δυνατότητα να βρούμε τον αντιπρόσωπο γρήγορα, προσθέτουμε στη δομή `Node` ένα ακόμα πεδίο `rep` που είναι αναφορά στον κόμβο-αντιπρόσωπο

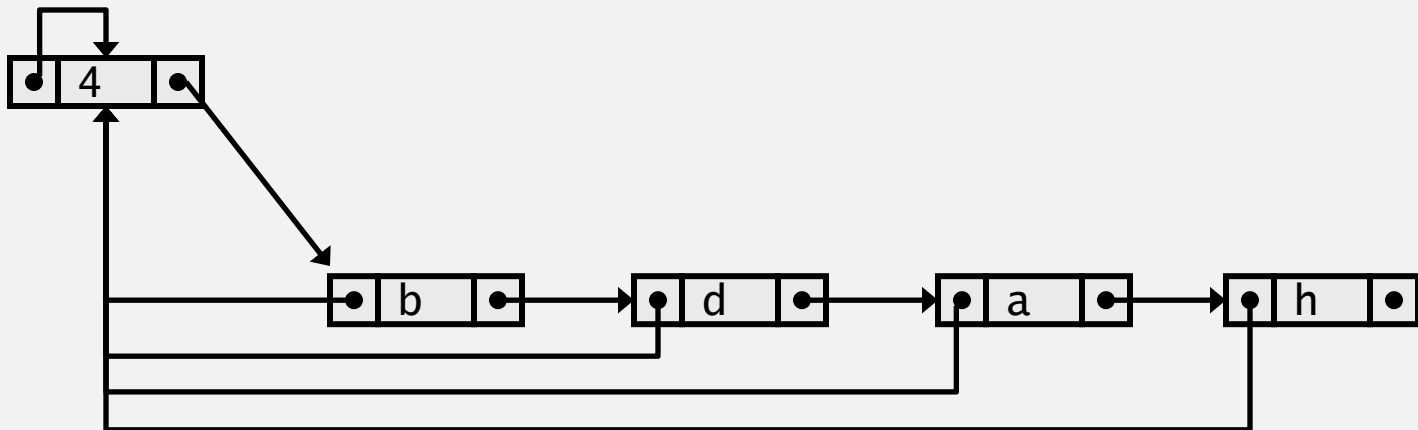
$unite(x, y)$: Έστω $p=x.rep$ και $q=y.rep$. Διατρέχει τη λίστα του q και για κάθε κόμβο v αυτής της λίστας θέτει $v.rep = p$. Τέλος συνενώνει τις δύο λίστες σε μία λίστα με αρχικό κόμβο το p .

Με λίγη περισσότερη προσπάθεια ο χρόνος της ένωσης γίνεται $O(\min\{n_x, n_y\})$ όπου n_x και n_y ο αριθμός των στοιχείων του συνόλου που περιέχει το x και το y αντίστοιχα

Δομή Εύρεσης-Ένωσης με Λίστες

Με λίγη περισσότερη προσπάθεια ο χρόνος της ένωσης γίνεται $O(\min\{n_x, n_y\})$ όπου n_x και n_y ο αριθμός των στοιχείων του συνόλου που περιέχει το x και το y αντίστοιχα

Τώρα πρέπει να αποθηκεύουμε και τον αριθμό των στοιχείων του κάθε συνόλου, π.χ. σε ξεχωριστό κόμβο-κεφαλή που παίζει το ρόλο του αντιπρόσωπου



Άσκηση: Δώστε τις λεπτομέρειες αυτής της δομής, ειδικά το πως υλοποιείται η *unite*

Δομή Εύρεσης-Ένωσης με Λίστες

Με λίγη περισσότερη προσπάθεια ο χρόνος της ένωσης γίνεται $O(\min\{n_x, n_y\})$ όπου n_x και n_y ο αριθμός των στοιχείων του συνόλου που περιέχει το x και το y αντίστοιχα

Αν ισχύει ο παραπάνω χρόνος για μια ένωση, τότε m ενώσεις γίνονται σε χρόνο $O(m + n \log n)$ όπου n ο συνολικός αριθμός των στοιχείων

Απόδειξη

Δομή Εύρεσης-Ένωσης με Λίστες

Με λίγη περισσότερη προσπάθεια ο χρόνος της ένωσης γίνεται $O(\min\{n_x, n_y\})$ όπου n_x και n_y ο αριθμός των στοιχείων του συνόλου που περιέχει το x και το y αντίστοιχα

Αν ισχύει ο παραπάνω χρόνος για μια ένωση, τότε m ενώσεις γίνονται σε χρόνο $O(m + n \log n)$ όπου n ο συνολικός αριθμός των στοιχείων

Απόδειξη

Έστω $n_x \geq n_y$. Τότε η $unite(x, y)$ εκτελείται σε χρόνο $O(n_y)$ και το νέο σύνολο που προκύπτει έχει $n_x + n_y \geq 2n_y$ στοιχεία. Το κόστος $O(n_y)$ επιμερίζεται σε κόστος $O(1)$ ανά στοιχείο που αλλάζει δείκτη σε αντιπρόσωπο.

Επομένως αν το y συμμετέχει στο μικρότερο σύνολο σε j ενώσεις τότε το νέο σύνολο που περιέχει το y θα έχει $\geq 2^j$ στοιχεία. Όμως $n \geq 2^j$ οπότε $j \leq \lg n$. Άρα το συνολικό κόστος ανά στοιχείο είναι $O(\log n)$.

Αφού έχουμε n στοιχεία το συνολικό κόστος είναι $O(m + n \log n)$

Δομή Εύρεσης-Ένωσης με Λίστες

Με λίγη περισσότερη προσπάθεια ο χρόνος της ένωσης γίνεται $O(\min\{n_x, n_y\})$ όπου n_x και n_y ο αριθμός των στοιχείων του συνόλου που περιέχει το x και το y αντίστοιχα

Αν ισχύει ο παραπάνω χρόνος για μια ένωση, τότε m ενώσεις γίνονται σε χρόνο $O(m + n \log n)$ όπου n ο συνολικός αριθμός των στοιχείων

Παρατήρηση

Ο παράγοντας $O(m)$ στο χρόνο $O(m + n \log n)$ αντιστοιχεί στο κόστος εύρεσης των αντιπροσώπων των συνόλων που ενώνονται

Με βάση τα παραπάνω καταλήγουμε στο συμπέρασμα ότι οποιαδήποτε ακολουθία από n λειτουργίες *makeset* και m λειτουργίες *unite* και *find* απαιτεί συνολικό χρόνο $O(m + n \log n)$

Δομή Εύρεσης-Ένωσης με Λίστες

Ερώτημα : Πως μπορούμε να βρούμε τον κόμβο που περιέχει ένα στοιχείο $k_i \in K$;

Αν τα στοιχεία $k_1, k_2, \dots, k_n$ είναι μικροί ακέραιοι (π.χ. $1, 2, \dots, n$) τότε μπορούμε να αποθηκεύσουμε ένα πίνακα δεικτών A όπου η θέση $A[k_i]$ δείχνει στον κόμβο που αντιστοιχεί στο στοιχείο k_i

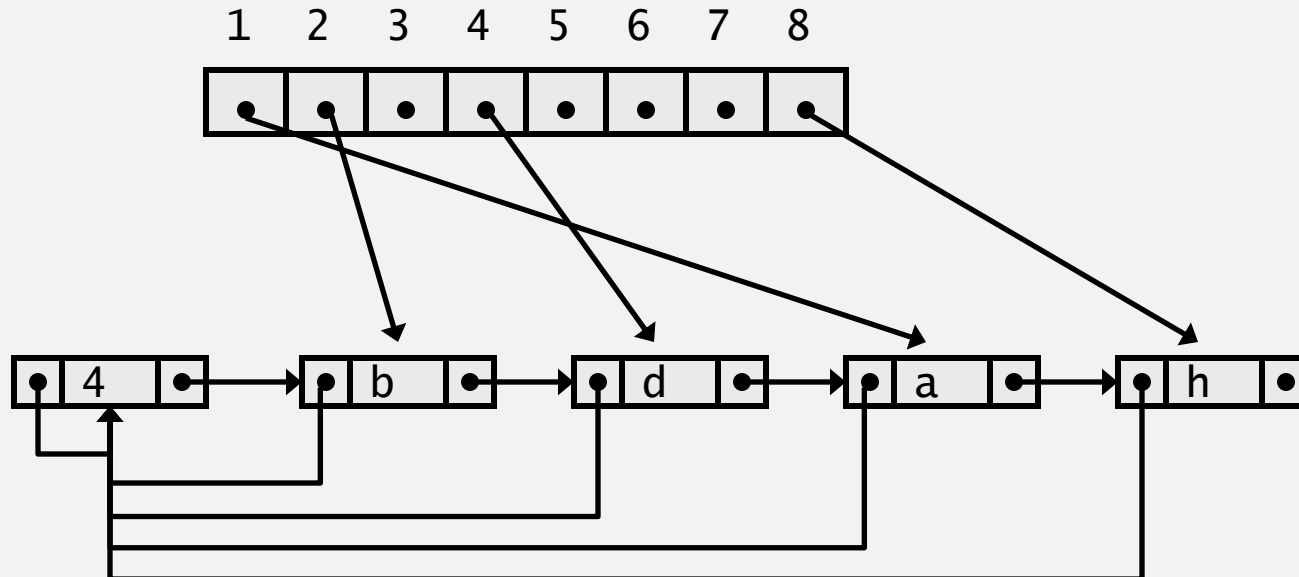
Π.χ.

$$a = 1$$

$$b = 2$$

$$d = 4$$

$$h = 8$$



Δομή Εύρεσης-Ένωσης με Λίστες

Ερώτημα : Πως μπορούμε να βρούμε τον κόμβο που περιέχει ένα στοιχείο $k_i \in K$;

Αν τα στοιχεία $k_1, k_2, \dots, k_n$ είναι μικροί ακέραιοι (π.χ. $1, 2, \dots, n$) τότε μπορούμε να αποθηκεύσουμε ένα πίνακα δεικτών A όπου η θέση $A[k_i]$ δείχνει στον κόμβο που αντιστοιχεί στο στοιχείο k_i

Διαφορετικά χρειαζόμαστε μια δομή αναζήτησης (λεξικό), π.χ. δένδρο αναζήτησης ή πίνακα διασποράς.

Αλγόριθμοι ένωσης-εύρεσης

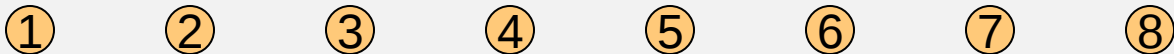
Ας υποθέσουμε ότι τα αντικείμενα μας είναι ακέραιοι αριθμοί: $A = \{1, 2, \dots, \nu\}$

Χρησιμοποιούμε ένα πίνακα Π διαστάσεων $1 \times \nu$

Αρχικοποίηση : $\Pi(\kappa) = \kappa$ για $\kappa = 1, 2, \dots, \nu$

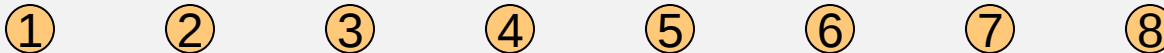
$$\Pi = \begin{bmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \end{bmatrix} \quad (\nu = 8)$$

Αρχικά κάθε αντικείμενο αποτελεί ένα ξεχωριστό σύνολο

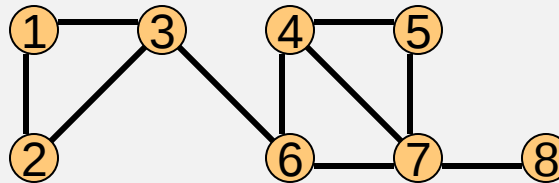


Αλγόριθμοι ένωσης-εύρεσης

Αρχικά κάθε αντικείμενο αποτελεί ένα ξεχωριστό σύνολο



Ο αλγόριθμος διαβάζει ένα γράφημα από την είσοδο και πραγματοποιεί τις κατάλληλες ενώσεις. Το γράφημα δίνεται ως μια ακολουθία ακμών.



Π.χ. το παραπάνω γράφημα μπορεί να μας δίνεται ως η ακολουθία ακμών

$\{1,2\} \{2,3\} \{1,3\} \{7,8\} \{4,6\} \{4,5\} \{6,7\} \{3,6\} \{5,7\} \{4,7\}$

Αλγόριθμοι ένωσης-εύρεσης: γρήγορη εύρεση

Επιβάλλουμε την παρακάτω συνθήκη:

Οποιαδήποτε δύο αντικείμενα α και β ανήκουν στο ίδιο σύνολο
αν και μόνο αν $\Pi(\alpha) = \Pi(\beta)$

$\text{ένωση}(\alpha, \beta)$: για κάθε κ τέτοιο ώστε $\Pi(\kappa) = \Pi(\beta)$ θέτουμε $\Pi(\kappa) \leftarrow \Pi(\alpha)$

$\text{εύρεση}(\alpha)$: επιστρέφουμε την τιμή του $\Pi(\alpha)$

Αλγόριθμοι ένωσης-εύρεσης: γρήγορη εύρεση

Οποιαδήποτε δύο αντικείμενα α και β ανήκουν στο ίδιο σύνολο
αν και μόνο αν $\Pi(\alpha) = \Pi(\beta)$

ένωση(α, β) : για κάθε κ τέτοιο ώστε $\Pi(\kappa) = \Pi(\beta)$ θέτουμε $\Pi(\kappa) \leftarrow \Pi(\alpha)$

```
void union(int a, int b)
{
    int k;
    if (Pi[a] == Pi[b]) return;
    for (k=1; k<=N; k++)
        if (Pi[k]==Pi[b]) Pi[k]=Pi[a];
}
```

εύρεση(α) : επιστρέφουμε την τιμή του $\Pi(\alpha)$

```
void find(int a)
{ return Pi[a]; }
```

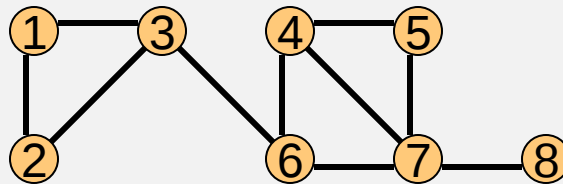
Αλγόριθμοι ένωσης-εύρεσης: γρήγορη εύρεση

ένωση(α, β) : για κάθε κ τέτοιο ώστε $\Pi(\kappa) = \Pi(\beta)$ θέτουμε $\Pi(\kappa) \leftarrow \Pi(\alpha)$

ακολουθία ακμών

$\{1,2\} \{2,3\} \{1,3\} \{7,8\} \{4,6\} \{4,5\} \{6,7\} \{3,6\} \{5,7\} \{4,7\}$

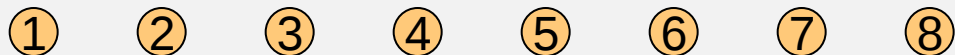
γράφημα



πίνακας

$\Pi = [\quad 1 \quad 2 \quad 3 \quad 4 \quad 5 \quad 6 \quad 7 \quad 8 \quad]$

δέντρο ένωσης-εύρεσης



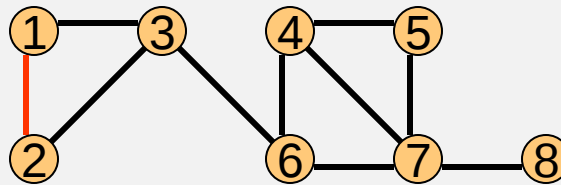
Αλγόριθμοι ένωσης-εύρεσης: γρήγορη εύρεση

ένωση(α, β) : για κάθε κ τέτοιο ώστε $\Pi(\kappa) = \Pi(\beta)$ θέτουμε $\Pi(\kappa) \leftarrow \Pi(\alpha)$

ακολουθία ακμών

$\{1,2\} \{2,3\} \{1,3\} \{7,8\} \{4,6\} \{4,5\} \{6,7\} \{3,6\} \{5,7\} \{4,7\}$

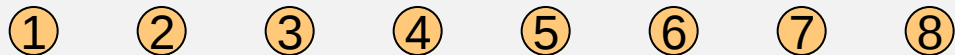
γράφημα



πίνακας

$\Pi = [\quad 1 \quad 2 \quad 3 \quad 4 \quad 5 \quad 6 \quad 7 \quad 8 \quad]$

δέντρο ένωσης-εύρεσης



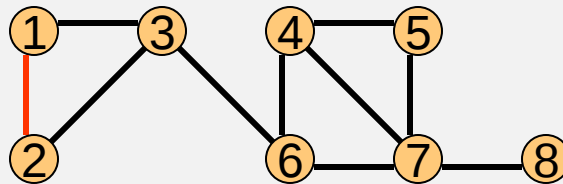
Αλγόριθμοι ένωσης-εύρεσης: γρήγορη εύρεση

ένωση(α, β) : για κάθε κ τέτοιο ώστε $\Pi(\kappa) = \Pi(\beta)$ θέτουμε $\Pi(\kappa) \leftarrow \Pi(\alpha)$

ακολουθία ακμών

$\{1,2\} \{2,3\} \{1,3\} \{7,8\} \{4,6\} \{4,5\} \{6,7\} \{3,6\} \{5,7\} \{4,7\}$

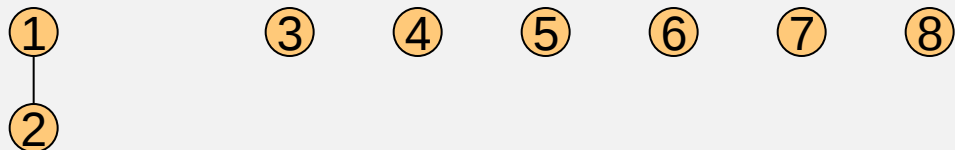
γράφημα



πίνακας

$\Pi = [1 \quad 1 \quad 3 \quad 4 \quad 5 \quad 6 \quad 7 \quad 8]$

δέντρο ένωσης-εύρεσης



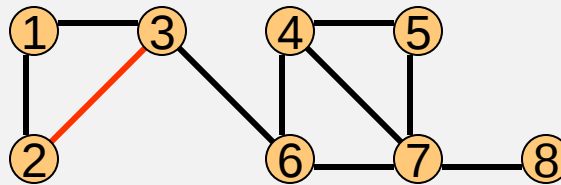
Αλγόριθμοι ένωσης-εύρεσης: γρήγορη εύρεση

ένωση(α, β) : για κάθε κ τέτοιο ώστε $\Pi(\kappa) = \Pi(\beta)$ θέτουμε $\Pi(\kappa) \leftarrow \Pi(\alpha)$

ακολουθία ακμών

$\{1,2\}$ $\{2,3\}$ $\{1,3\}$ $\{7,8\}$ $\{4,6\}$ $\{4,5\}$ $\{6,7\}$ $\{3,6\}$ $\{5,7\}$ $\{4,7\}$

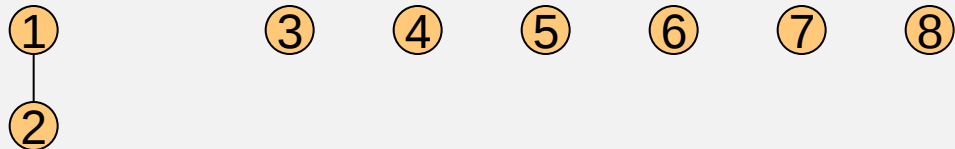
γράφημα



πίνακας

$\Pi = [\quad 1 \quad 1 \quad 3 \quad 4 \quad 5 \quad 6 \quad 7 \quad 8 \quad]$

δέντρο ένωσης-εύρεσης



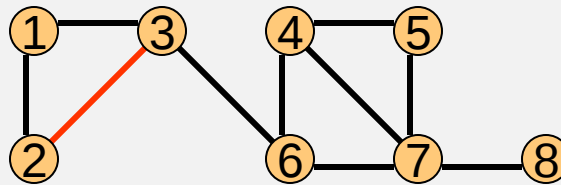
Αλγόριθμοι ένωσης-εύρεσης: γρήγορη εύρεση

ένωση(α, β) : για κάθε κ τέτοιο ώστε $\Pi(\kappa) = \Pi(\beta)$ θέτουμε $\Pi(\kappa) \leftarrow \Pi(\alpha)$

ακολουθία ακμών

$\{1,2\}$ $\{2,3\}$ $\{1,3\}$ $\{7,8\}$ $\{4,6\}$ $\{4,5\}$ $\{6,7\}$ $\{3,6\}$ $\{5,7\}$ $\{4,7\}$

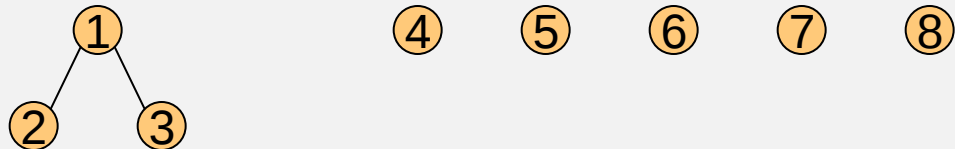
γράφημα



πίνακας

$\Pi = [1 \quad 1 \quad 1 \quad 4 \quad 5 \quad 6 \quad 7 \quad 8]$

δέντρο ένωσης-εύρεσης



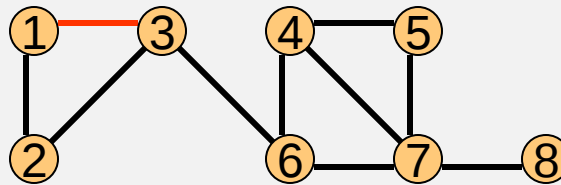
Αλγόριθμοι ένωσης-εύρεσης: γρήγορη εύρεση

ένωση(α, β) : για κάθε κ τέτοιο ώστε $\Pi(\kappa) = \Pi(\beta)$ θέτουμε $\Pi(\kappa) \leftarrow \Pi(\alpha)$

ακολουθία ακμών

$\{1,2\} \{2,3\} \{1,3\} \{7,8\} \{4,6\} \{4,5\} \{6,7\} \{3,6\} \{5,7\} \{4,7\}$

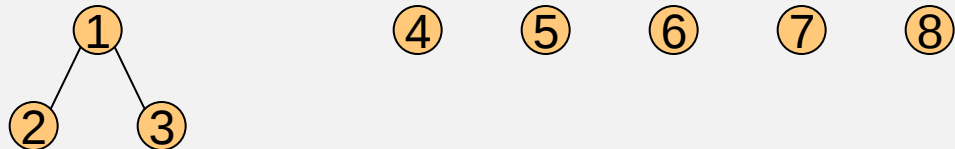
γράφημα



πίνακας

$\Pi = [1 \quad 1 \quad 1 \quad 4 \quad 5 \quad 6 \quad 7 \quad 8]$

δέντρο ένωσης-εύρεσης



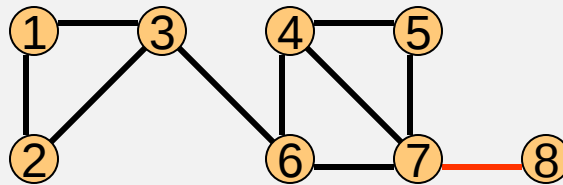
Αλγόριθμοι ένωσης-εύρεσης: γρήγορη εύρεση

ένωση(α, β) : για κάθε κ τέτοιο ώστε $\Pi(\kappa) = \Pi(\beta)$ θέτουμε $\Pi(\kappa) \leftarrow \Pi(\alpha)$

ακολουθία ακμών

$\{1,2\} \{2,3\} \{1,3\} \{7,8\} \{4,6\} \{4,5\} \{6,7\} \{3,6\} \{5,7\} \{4,7\}$

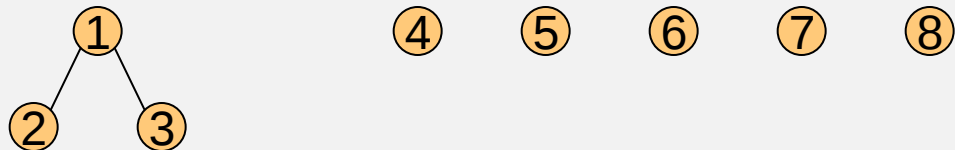
γράφημα



πίνακας

$\Pi = [1 \quad 1 \quad 1 \quad 4 \quad 5 \quad 6 \quad 7 \quad 8]$

δέντρο ένωσης-εύρεσης



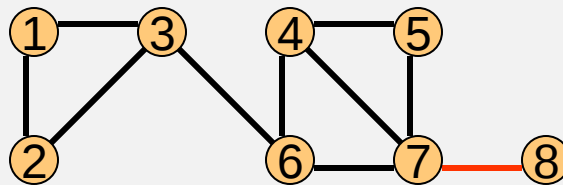
Αλγόριθμοι ένωσης-εύρεσης: γρήγορη εύρεση

ένωση(α, β) : για κάθε κ τέτοιο ώστε $\Pi(\kappa) = \Pi(\beta)$ θέτουμε $\Pi(\kappa) \leftarrow \Pi(\alpha)$

ακολουθία ακμών

$\{1,2\} \{2,3\} \{1,3\} \{7,8\} \{4,6\} \{4,5\} \{6,7\} \{3,6\} \{5,7\} \{4,7\}$

γράφημα



πίνακας

$\Pi = [1 \quad 1 \quad 1 \quad 4 \quad 5 \quad 6 \quad 7 \quad 7]$

δέντρο ένωσης-εύρεσης



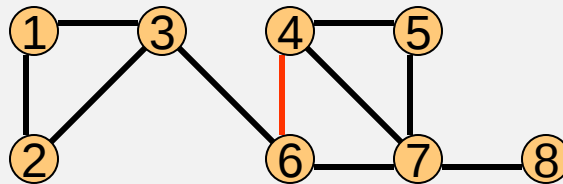
Αλγόριθμοι ένωσης-εύρεσης: γρήγορη εύρεση

ένωση(α, β) : για κάθε κ τέτοιο ώστε $\Pi(\kappa) = \Pi(\beta)$ θέτουμε $\Pi(\kappa) \leftarrow \Pi(\alpha)$

ακολουθία ακμών

$\{1,2\} \{2,3\} \{1,3\} \{7,8\} \{4,6\} \{4,5\} \{6,7\} \{3,6\} \{5,7\} \{4,7\}$

γράφημα



πίνακας

$\Pi = [1 \quad 1 \quad 1 \quad 4 \quad 5 \quad 6 \quad 7 \quad 7]$

δέντρο ένωσης-εύρεσης



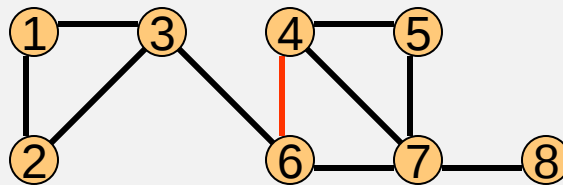
Αλγόριθμοι ένωσης-εύρεσης: γρήγορη εύρεση

ένωση(α, β) : για κάθε κ τέτοιο ώστε $\Pi(\kappa) = \Pi(\beta)$ θέτουμε $\Pi(\kappa) \leftarrow \Pi(\alpha)$

ακολουθία ακμών

$\{1,2\} \{2,3\} \{1,3\} \{7,8\} \{4,6\} \{4,5\} \{6,7\} \{3,6\} \{5,7\} \{4,7\}$

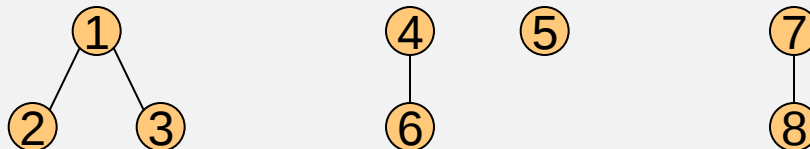
γράφημα



πίνακας

$\Pi = [1 \quad 1 \quad 1 \quad 4 \quad 5 \quad 4 \quad 7 \quad 7]$

δέντρο ένωσης-εύρεσης



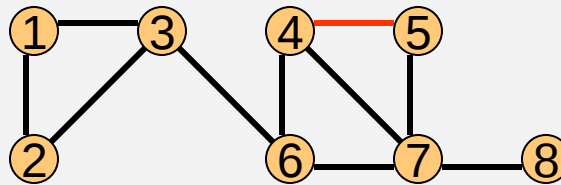
Αλγόριθμοι ένωσης-εύρεσης: γρήγορη εύρεση

ένωση(α, β) : για κάθε κ τέτοιο ώστε $\Pi(\kappa) = \Pi(\beta)$ θέτουμε $\Pi(\kappa) \leftarrow \Pi(\alpha)$

ακολουθία ακμών

$\{1,2\} \{2,3\} \{1,3\} \{7,8\} \{4,6\} \{4,5\} \{6,7\} \{3,6\} \{5,7\} \{4,7\}$

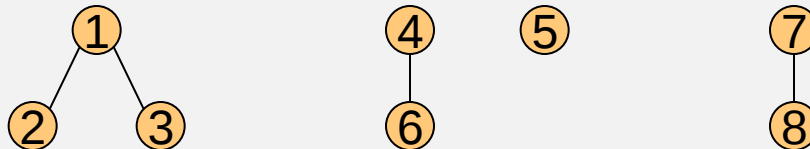
γράφημα



πίνακας

$\Pi = [1 \quad 1 \quad 1 \quad 4 \quad 5 \quad 4 \quad 7 \quad 7]$

δέντρο ένωσης-εύρεσης



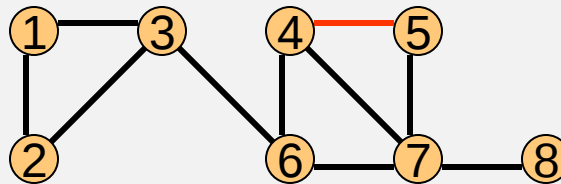
Αλγόριθμοι ένωσης-εύρεσης: γρήγορη εύρεση

ένωση(α, β) : για κάθε κ τέτοιο ώστε $\Pi(\kappa) = \Pi(\beta)$ θέτουμε $\Pi(\kappa) \leftarrow \Pi(\alpha)$

ακολουθία ακμών

$\{1,2\} \{2,3\} \{1,3\} \{7,8\} \{4,6\} \{4,5\} \{6,7\} \{3,6\} \{5,7\} \{4,7\}$

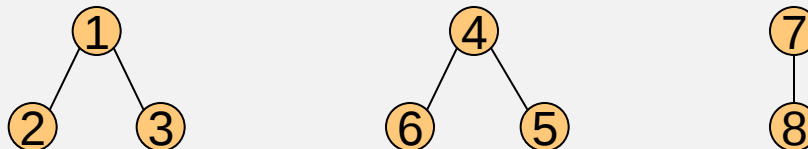
γράφημα



πίνακας

$\Pi = [1 \quad 1 \quad 1 \quad 4 \quad 4 \quad 4 \quad 7 \quad 7]$

δέντρο ένωσης-εύρεσης



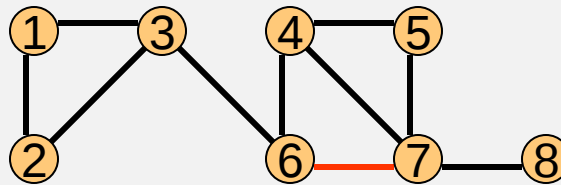
Αλγόριθμοι ένωσης-εύρεσης: γρήγορη εύρεση

ένωση(α, β) : για κάθε κ τέτοιο ώστε $\Pi(\kappa) = \Pi(\beta)$ θέτουμε $\Pi(\kappa) \leftarrow \Pi(\alpha)$

ακολουθία ακμών

$\{1,2\} \{2,3\} \{1,3\} \{7,8\} \{4,6\} \{4,5\} \{6,7\} \{3,6\} \{5,7\} \{4,7\}$

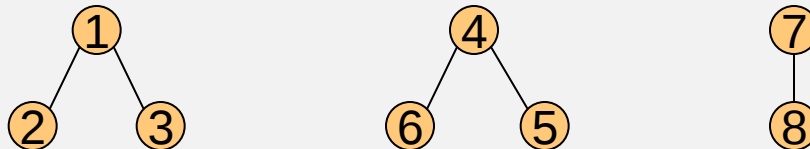
γράφημα



πίνακας

$\Pi = [1 \quad 1 \quad 1 \quad 4 \quad 4 \quad 4 \quad 7 \quad 7]$

δέντρο ένωσης-εύρεσης



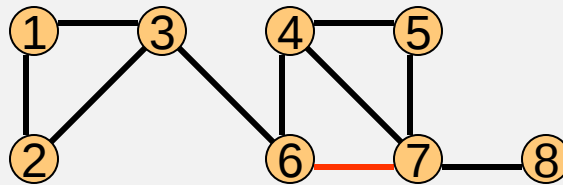
Αλγόριθμοι ένωσης-εύρεσης: γρήγορη εύρεση

ένωση(α, β) : για κάθε κ τέτοιο ώστε $\Pi(\kappa) = \Pi(\beta)$ θέτουμε $\Pi(\kappa) \leftarrow \Pi(\alpha)$

ακολουθία ακμών

$\{1,2\} \{2,3\} \{1,3\} \{7,8\} \{4,6\} \{4,5\} \{6,7\} \{3,6\} \{5,7\} \{4,7\}$

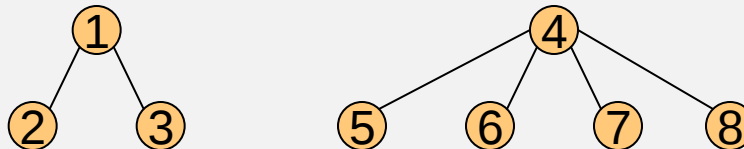
γράφημα



πίνακας

$$\Pi = \begin{bmatrix} 1 & 1 & 1 & 4 & 4 & 4 & 4 & 4 \end{bmatrix}$$

δέντρο ένωσης-εύρεσης



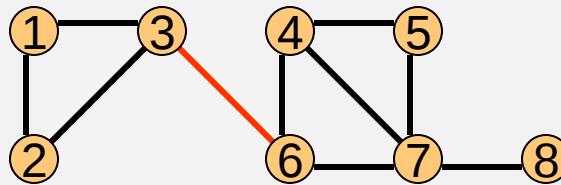
Αλγόριθμοι ένωσης-εύρεσης: γρήγορη εύρεση

ένωση(α, β) : για κάθε κ τέτοιο ώστε $\Pi(\kappa) = \Pi(\beta)$ θέτουμε $\Pi(\kappa) \leftarrow \Pi(\alpha)$

ακολουθία ακμών

$\{1,2\} \{2,3\} \{1,3\} \{7,8\} \{4,6\} \{4,5\} \{6,7\} \{3,6\} \{5,7\} \{4,7\}$

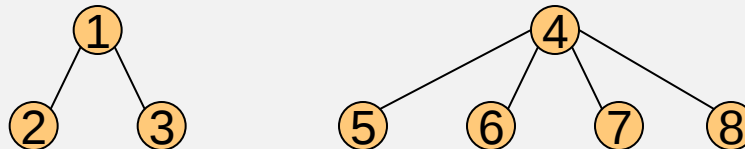
γράφημα



πίνακας

$$\Pi = \begin{bmatrix} 1 & 1 & 1 & 4 & 4 & 4 & 4 & 4 \end{bmatrix}$$

δέντρο ένωσης-εύρεσης



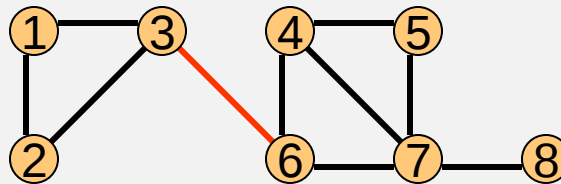
Αλγόριθμοι ένωσης-εύρεσης: γρήγορη εύρεση

ένωση(α, β) : για κάθε κ τέτοιο ώστε $\Pi(\kappa) = \Pi(\beta)$ θέτουμε $\Pi(\kappa) \leftarrow \Pi(\alpha)$

ακολουθία ακμών

$\{1,2\} \{2,3\} \{1,3\} \{7,8\} \{4,6\} \{4,5\} \{6,7\} \{3,6\} \{5,7\} \{4,7\}$

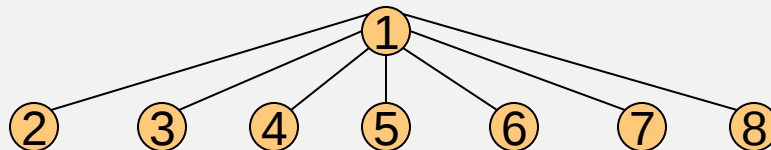
γράφημα



πίνακας

$$\Pi = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

δέντρο ένωσης-εύρεσης



Αλγόριθμοι ένωσης-εύρεσης: γρήγορη εύρεση

Επιβάλλουμε την παρακάτω συνθήκη:

Οποιαδήποτε δύο αντικείμενα α και β ανήκουν στο ίδιο σύνολο
αν και μόνο αν $\Pi(\alpha) = \Pi(\beta)$

ένωση(α, β) : για κάθε κ τέτοιο ώστε $\Pi(\kappa) = \Pi(\beta)$ θέτουμε $\Pi(\kappa) \leftarrow \Pi(\alpha)$

εύρεση(α) : επιστρέφουμε την τιμή του $\Pi(\alpha)$

Ιδιότητα: Για n αντικείμενα και m πράξεις ένωσης, ο αλγόριθμος γρήγορης εύρεσης απαιτεί τουλάχιστον μn εντολές.

Αλγόριθμοι ένωσης-εύρεσης: γρήγορη ένωση

$\Pi(\kappa)$: δείκτης σε άλλο αντικείμενο του ίδιου συνόλου

εύρεση(α) : ξεκινώντας από το α ακολουθούμε τους δείκτες μέχρι να βρούμε βρούμε αντικείμενο κ τέτοιο ώστε $\Pi(\kappa) = \kappa$ · επιστρέφουμε την τιμή κ .

α και β ανήκουν στο ίδιο σύνολο αν και μόνο αν $\text{εύρεση}(\alpha) = \text{εύρεση}(\beta)$

ένωση(α, β) : εκτελούμε $\kappa \leftarrow \text{εύρεση}(\alpha)$ και $\lambda \leftarrow \text{εύρεση}(\beta)$ · θέτουμε $\Pi(\lambda) = \kappa$

Αλγόριθμοι ένωσης-εύρεσης: γρήγορη ένωση

$\Pi(\kappa)$: δείκτης σε άλλο αντικείμενο του ίδιου συνόλου

εύρεση(α) : ξεκινώντας από το α ακολουθούμε τους δείκτες μέχρι να βρούμε βρούμε αντικείμενο κ τέτοιο ώστε $\Pi(\kappa) = \kappa$ · επιστρέφουμε την τιμή κ .

```
void find(int a)
{
    int k=a;
    while (k != Pi[k]) k=Pi[k];
    return k;
}
```

α και β ανήκουν στο ίδιο σύνολο αν και μόνο αν $\text{εύρεση}(\alpha) = \text{εύρεση}(\beta)$

ένωση(α, β) : εκτελούμε $\kappa \leftarrow \text{εύρεση}(\alpha)$ και $\lambda \leftarrow \text{εύρεση}(\beta)$ · θέτουμε $\Pi(\lambda) = \kappa$

```
void union(int a, int b)
{
    int k=find(a);
    int l=find(b);
    Pi[l]=k;
}
```

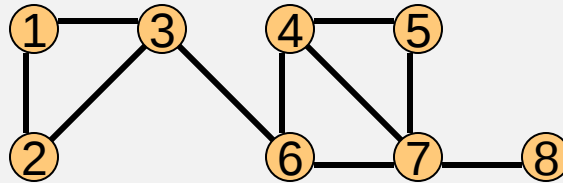
Αλγόριθμοι ένωσης-εύρεσης: γρήγορη ένωση

ένωση(α, β) : εκτελούμε $\kappa \leftarrow \text{εύρεση}(\alpha)$ και $\lambda \leftarrow \text{εύρεση}(\beta)$ · θέτουμε $\Pi(\lambda) = \kappa$

ακολουθία ακμών

$\{1,2\} \{2,3\} \{1,3\} \{7,8\} \{4,6\} \{4,5\} \{6,7\} \{3,6\} \{5,7\} \{4,7\}$

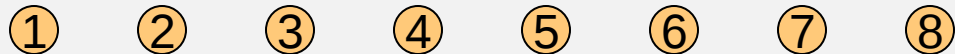
γράφημα



πίνακας

$\Pi = [\quad 1 \quad 2 \quad 3 \quad 4 \quad 5 \quad 6 \quad 7 \quad 8 \quad]$

δέντρο ένωσης-εύρεσης



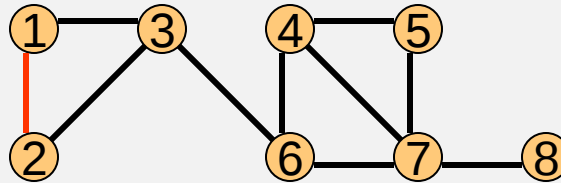
Αλγόριθμοι ένωσης-εύρεσης: γρήγορη ένωση

ένωση(α, β) : εκτελούμε $\kappa \leftarrow \text{εύρεση}(\alpha)$ και $\lambda \leftarrow \text{εύρεση}(\beta)$ · θέτουμε $\Pi(\lambda) = \kappa$

ακολουθία ακμών

$\{1,2\} \{2,3\} \{1,3\} \{7,8\} \{4,6\} \{4,5\} \{6,7\} \{3,6\} \{5,7\} \{4,7\}$

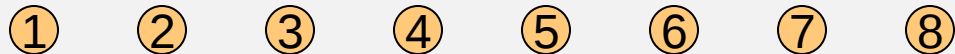
γράφημα



πίνακας

$\Pi = [\quad 1 \quad 2 \quad 3 \quad 4 \quad 5 \quad 6 \quad 7 \quad 8 \quad]$

δέντρο ένωσης-εύρεσης



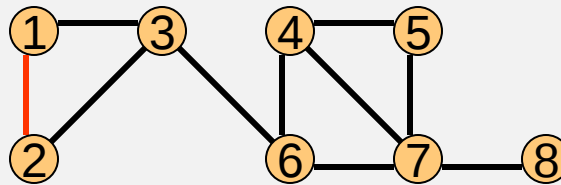
Αλγόριθμοι ένωσης-εύρεσης: γρήγορη ένωση

ένωση(α, β) : εκτελούμε $\kappa \leftarrow \text{εύρεση}(\alpha)$ και $\lambda \leftarrow \text{εύρεση}(\beta)$ · θέτουμε $\Pi(\lambda) = \kappa$

ακολουθία ακμών

$\{1,2\} \{2,3\} \{1,3\} \{7,8\} \{4,6\} \{4,5\} \{6,7\} \{3,6\} \{5,7\} \{4,7\}$

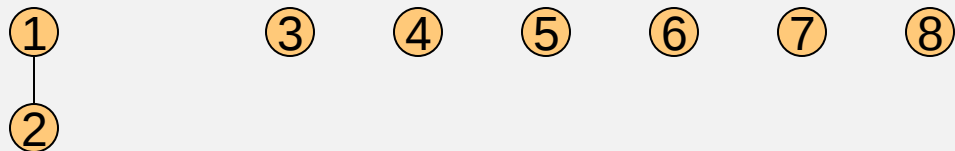
γράφημα



πίνακας

$\Pi = [1 \quad 1 \quad 3 \quad 4 \quad 5 \quad 6 \quad 7 \quad 8]$

δέντρο ένωσης-εύρεσης



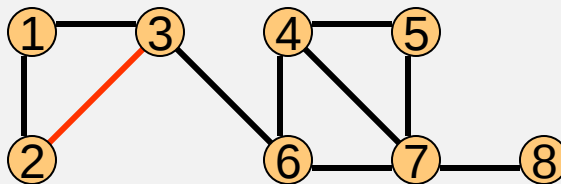
Αλγόριθμοι ένωσης-εύρεσης: γρήγορη ένωση

ένωση(α, β) : εκτελούμε $\kappa \leftarrow \text{εύρεση}(\alpha)$ και $\lambda \leftarrow \text{εύρεση}(\beta)$ · θέτουμε $\Pi(\lambda) = \kappa$

ακολουθία ακμών

$\{1,2\}$ $\{2,3\}$ $\{1,3\}$ $\{7,8\}$ $\{4,6\}$ $\{4,5\}$ $\{6,7\}$ $\{3,6\}$ $\{5,7\}$ $\{4,7\}$

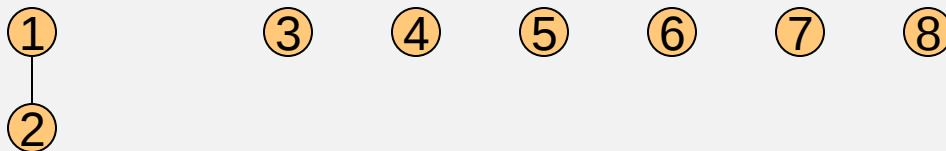
γράφημα



πίνακας

$\Pi = [\quad 1 \quad 1 \quad 3 \quad 4 \quad 5 \quad 6 \quad 7 \quad 8 \quad]$

δέντρο ένωσης-εύρεσης



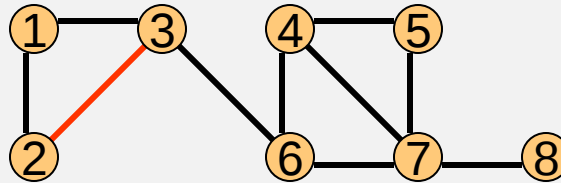
Αλγόριθμοι ένωσης-εύρεσης: γρήγορη ένωση

ένωση(α, β) : εκτελούμε $\kappa \leftarrow \text{εύρεση}(\alpha)$ και $\lambda \leftarrow \text{εύρεση}(\beta)$ · θέτουμε $\Pi(\lambda) = \kappa$

ακολουθία ακμών

$\{1,2\}$ $\{2,3\}$ $\{1,3\}$ $\{7,8\}$ $\{4,6\}$ $\{4,5\}$ $\{6,7\}$ $\{3,6\}$ $\{5,7\}$ $\{4,7\}$

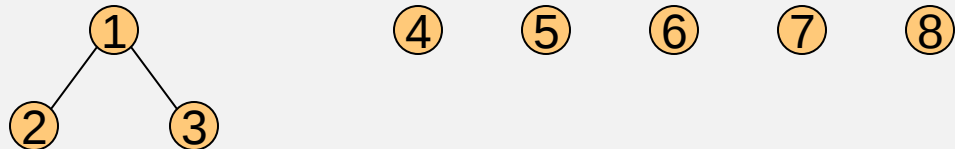
γράφημα



πίνακας

$\Pi = [1 \quad 1 \quad 1 \quad 4 \quad 5 \quad 6 \quad 7 \quad 8]$

δέντρο ένωσης-εύρεσης



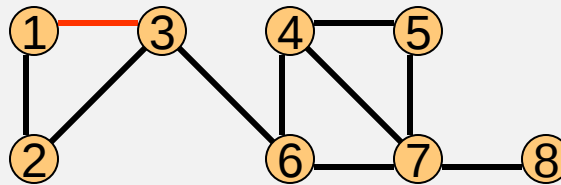
Αλγόριθμοι ένωσης-εύρεσης: γρήγορη ένωση

ένωση(α, β) : εκτελούμε $\kappa \leftarrow \text{εύρεση}(\alpha)$ και $\lambda \leftarrow \text{εύρεση}(\beta)$ · θέτουμε $\Pi(\lambda) = \kappa$

ακολουθία ακμών

{1,2} {2,3} {1,3} {7,8} {4,6} {4,5} {6,7} {3,6} {5,7} {4,7}

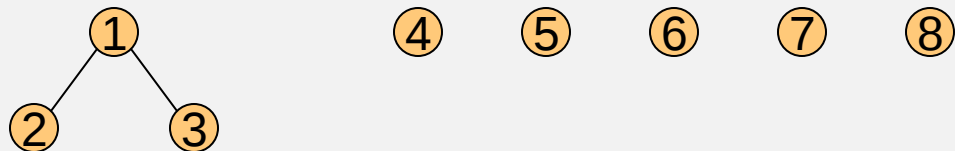
γράφημα



πίνακας

$\Pi = [1 \quad 1 \quad 1 \quad 4 \quad 5 \quad 6 \quad 7 \quad 8]$

δέντρο ένωσης-εύρεσης



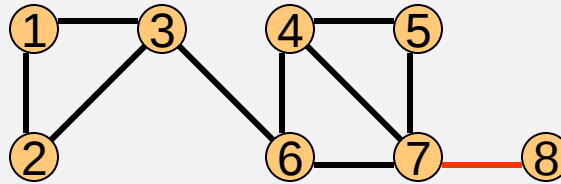
Αλγόριθμοι ένωσης-εύρεσης: γρήγορη ένωση

ένωση(α, β) : εκτελούμε $\kappa \leftarrow \text{εύρεση}(\alpha)$ και $\lambda \leftarrow \text{εύρεση}(\beta)$ · θέτουμε $\Pi(\lambda) = \kappa$

ακολουθία ακμών

$\{1,2\} \{2,3\} \{1,3\} \{7,8\} \{4,6\} \{4,5\} \{6,7\} \{3,6\} \{5,7\} \{4,7\}$

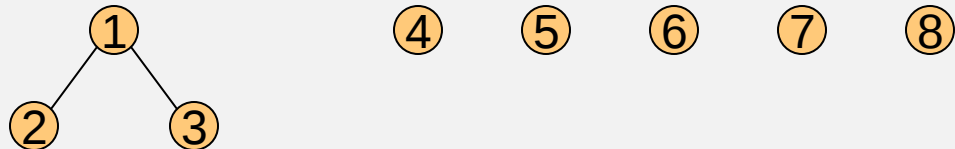
γράφημα



πίνακας

$\Pi = [1 \quad 1 \quad 1 \quad 4 \quad 5 \quad 6 \quad 7 \quad 8]$

δέντρο ένωσης-εύρεσης



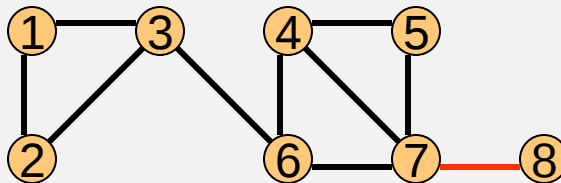
Αλγόριθμοι ένωσης-εύρεσης: γρήγορη ένωση

ένωση(α, β) : εκτελούμε $\kappa \leftarrow \text{εύρεση}(\alpha)$ και $\lambda \leftarrow \text{εύρεση}(\beta)$ · θέτουμε $\Pi(\lambda) = \kappa$

ακολουθία ακμών

$\{1,2\} \{2,3\} \{1,3\} \{7,8\} \{4,6\} \{4,5\} \{6,7\} \{3,6\} \{5,7\} \{4,7\}$

γράφημα



πίνακας

$\Pi = [1 \quad 1 \quad 1 \quad 4 \quad 5 \quad 6 \quad 7 \quad 7]$

δέντρο ένωσης-εύρεσης



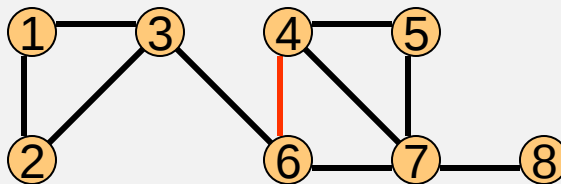
Αλγόριθμοι ένωσης-εύρεσης: γρήγορη ένωση

ένωση(α, β) : εκτελούμε $\kappa \leftarrow \text{εύρεση}(\alpha)$ και $\lambda \leftarrow \text{εύρεση}(\beta)$ · θέτουμε $\Pi(\lambda) = \kappa$

ακολουθία ακμών

$\{1,2\} \{2,3\} \{1,3\} \{7,8\} \{4,6\} \{4,5\} \{6,7\} \{3,6\} \{5,7\} \{4,7\}$

γράφημα



πίνακας

$\Pi = [1 \quad 1 \quad 1 \quad 4 \quad 5 \quad 6 \quad 7 \quad 7]$

δέντρο ένωσης-εύρεσης



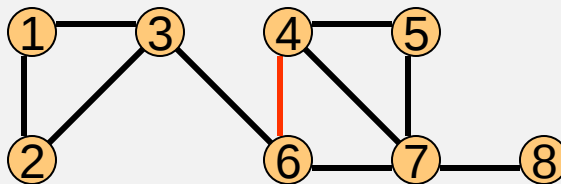
Αλγόριθμοι ένωσης-εύρεσης: γρήγορη ένωση

ένωση(α, β) : εκτελούμε $\kappa \leftarrow \text{εύρεση}(\alpha)$ και $\lambda \leftarrow \text{εύρεση}(\beta)$ · θέτουμε $\Pi(\lambda) = \kappa$

ακολουθία ακμών

$\{1,2\} \{2,3\} \{1,3\} \{7,8\} \{4,6\} \{4,5\} \{6,7\} \{3,6\} \{5,7\} \{4,7\}$

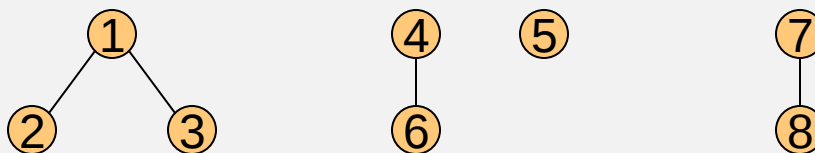
γράφημα



πίνακας

$\Pi = \begin{bmatrix} 1 & 1 & 1 & 4 & 5 & 4 & 7 & 7 \end{bmatrix}$

δέντρο ένωσης-εύρεσης



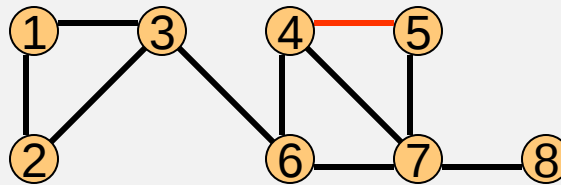
Αλγόριθμοι ένωσης-εύρεσης: γρήγορη ένωση

ένωση(α, β) : εκτελούμε $\kappa \leftarrow \text{εύρεση}(\alpha)$ και $\lambda \leftarrow \text{εύρεση}(\beta)$ · θέτουμε $\Pi(\lambda) = \kappa$

ακολουθία ακμών

$\{1,2\} \{2,3\} \{1,3\} \{7,8\} \{4,6\} \{4,5\} \{6,7\} \{3,6\} \{5,7\} \{4,7\}$

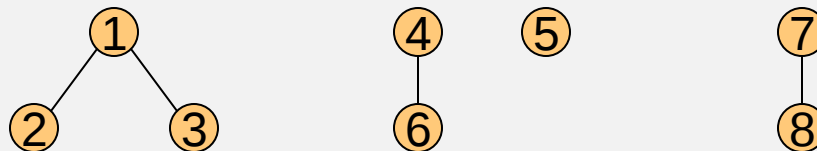
γράφημα



πίνακας

$\Pi = \begin{bmatrix} 1 & 1 & 1 & 4 & 5 & 4 & 7 & 7 \end{bmatrix}$

δέντρο ένωσης-εύρεσης



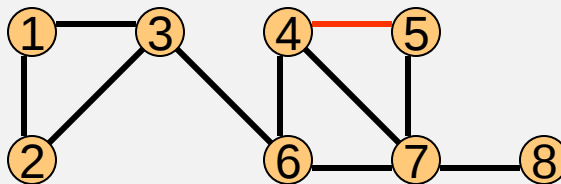
Αλγόριθμοι ένωσης-εύρεσης: γρήγορη ένωση

ένωση(α, β) : εκτελούμε $\kappa \leftarrow \text{εύρεση}(\alpha)$ και $\lambda \leftarrow \text{εύρεση}(\beta)$ · θέτουμε $\Pi(\lambda) = \kappa$

ακολουθία ακμών

$\{1,2\} \{2,3\} \{1,3\} \{7,8\} \{4,6\} \{4,5\} \{6,7\} \{3,6\} \{5,7\} \{4,7\}$

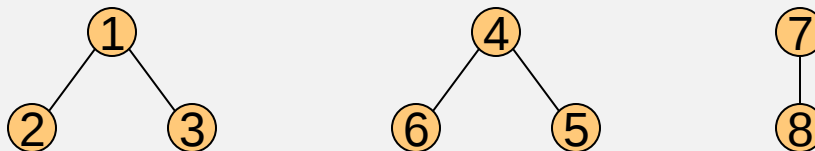
γράφημα



πίνακας

$\Pi = \begin{bmatrix} 1 & 1 & 1 & 4 & 4 & 4 & 7 & 7 \end{bmatrix}$

δέντρο ένωσης-εύρεσης



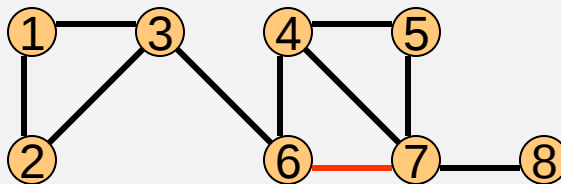
Αλγόριθμοι ένωσης-εύρεσης: γρήγορη ένωση

ένωση(α, β) : εκτελούμε $\kappa \leftarrow \text{εύρεση}(\alpha)$ και $\lambda \leftarrow \text{εύρεση}(\beta)$ · θέτουμε $\Pi(\lambda) = \kappa$

ακολουθία ακμών

$\{1,2\} \{2,3\} \{1,3\} \{7,8\} \{4,6\} \{4,5\} \{6,7\} \{3,6\} \{5,7\} \{4,7\}$

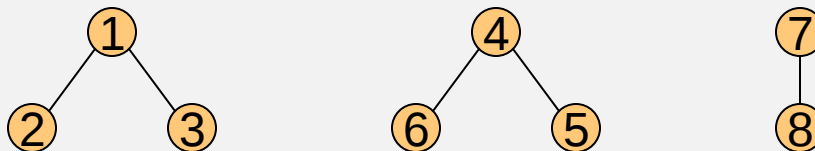
γράφημα



πίνακας

$\Pi = \begin{bmatrix} 1 & 1 & 1 & 4 & 4 & 4 & 7 & 7 \end{bmatrix}$

δέντρο ένωσης-εύρεσης



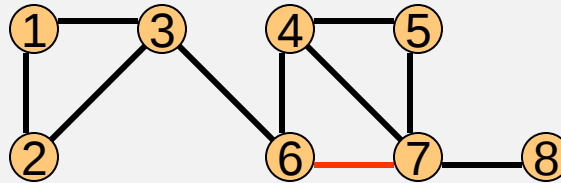
Αλγόριθμοι ένωσης-εύρεσης: γρήγορη ένωση

ένωση(α, β) : εκτελούμε $\kappa \leftarrow \text{εύρεση}(\alpha)$ και $\lambda \leftarrow \text{εύρεση}(\beta)$ · θέτουμε $\Pi(\lambda) = \kappa$

ακολουθία ακμών

$\{1,2\} \{2,3\} \{1,3\} \{7,8\} \{4,6\} \{4,5\} \{6,7\} \{3,6\} \{5,7\} \{4,7\}$

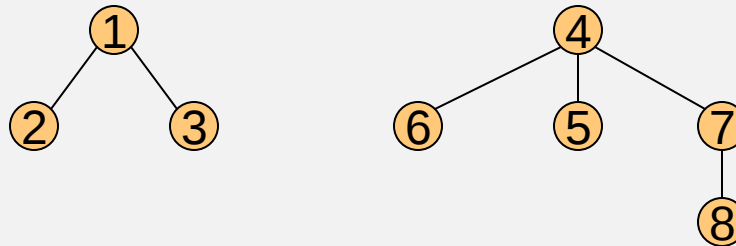
γράφημα



πίνακας

$\Pi = [1 \quad 1 \quad 1 \quad 4 \quad 4 \quad 4 \quad 4 \quad 7]$

δέντρο ένωσης-εύρεσης



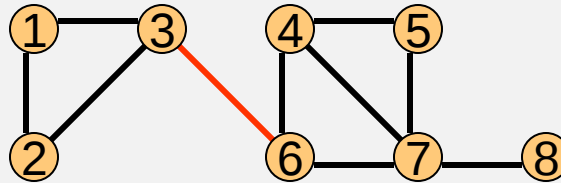
Αλγόριθμοι ένωσης-εύρεσης: γρήγορη ένωση

ένωση(α, β) : εκτελούμε $\kappa \leftarrow \text{εύρεση}(\alpha)$ και $\lambda \leftarrow \text{εύρεση}(\beta)$ · θέτουμε $\Pi(\lambda) = \kappa$

ακολουθία ακμών

{1,2} {2,3} {1,3} {7,8} {4,6} {4,5} {6,7} {3,6} {5,7} {4,7}

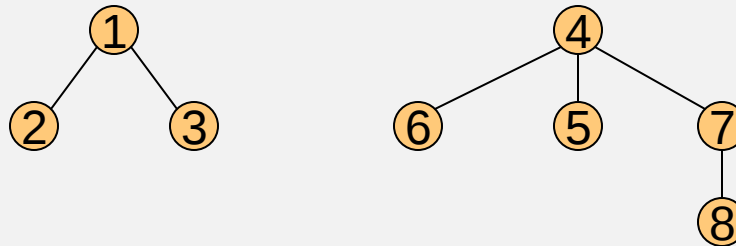
γράφημα



πίνακας

$\Pi = [1 \quad 1 \quad 1 \quad 4 \quad 4 \quad 4 \quad 4 \quad 7]$

δέντρο ένωσης-εύρεσης



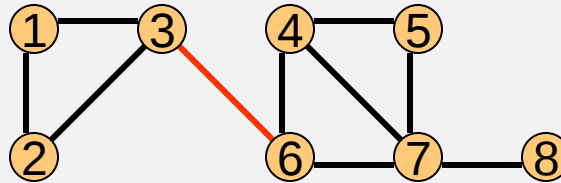
Αλγόριθμοι ένωσης-εύρεσης: γρήγορη ένωση

ένωση(α, β) : εκτελούμε $\kappa \leftarrow \text{εύρεση}(\alpha)$ και $\lambda \leftarrow \text{εύρεση}(\beta)$ · θέτουμε $\Pi(\lambda) = \kappa$

ακολουθία ακμών

$\{1,2\} \{2,3\} \{1,3\} \{7,8\} \{4,6\} \{4,5\} \{6,7\} \{3,6\} \{5,7\} \{4,7\}$

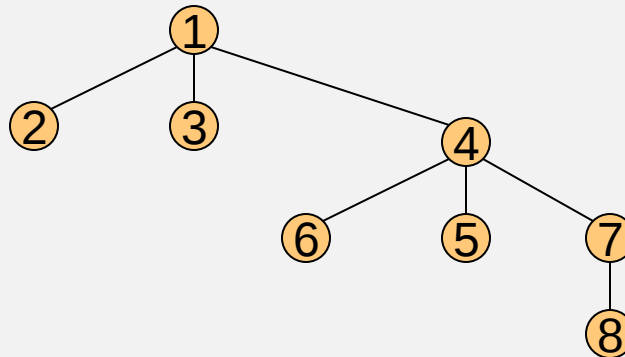
γράφημα



πίνακας

$\Pi = \begin{bmatrix} 1 & 1 & 1 & 1 & 4 & 4 & 4 & 7 \end{bmatrix}$

δέντρο ένωσης-εύρεσης



Αλγόριθμοι ένωσης-εύρεσης: γρήγορη ένωση

εύρεση(α) : ξεκινώντας από το α ακολουθούμε τους δείκτες μέχρι να βρούμε βρούμε αντικείμενο κ τέτοιο ώστε $\Pi(\kappa) = \kappa$ · επιστρέφουμε την τιμή κ .

ένωση(α, β) : εκτελούμε $\kappa \leftarrow \text{εύρεση}(\alpha)$ και $\lambda \leftarrow \text{εύρεση}(\beta)$ · θέτουμε $\Pi(\lambda) = \kappa$

Ιδιότητα: Για n αντικείμενα και $m \geq n$ πράξεις ένωσης, ο αλγόριθμος γρήγορης ένωσης μπορεί να χρειαστεί περισσότερες από $m \cdot n / 2$ εντολές.

Αλγόριθμοι ένωσης-εύρεσης: γρήγορη ένωση

εύρεση(α) : ξεκινώντας από το α ακολουθούμε τους δείκτες μέχρι να βρούμε βρούμε αντικείμενο κ τέτοιο ώστε $\Pi(\kappa) = \kappa$ · επιστρέφουμε την τιμή κ .

ένωση(α, β) : εκτελούμε $\kappa \leftarrow \text{εύρεση}(\alpha)$ και $\lambda \leftarrow \text{εύρεση}(\beta)$ · θέτουμε $\Pi(\lambda) = \kappa$

Ιδιότητα: Για n αντικείμενα και $m \geq n$ πράξεις ένωσης, ο αλγόριθμος γρήγορης ένωσης μπορεί να χρειαστεί περισσότερες από $m \cdot n / 2$ εντολές.

Απόδειξη: Η ακολουθία $\text{ένωση}(n-1, n)$,

7

8

Αλγόριθμοι ένωσης-εύρεσης: γρήγορη ένωση

εύρεση(α) : ξεκινώντας από το α ακολουθούμε τους δείκτες μέχρι να βρούμε βρούμε αντικείμενο κ τέτοιο ώστε $\Pi(\kappa) = \kappa$ · επιστρέφουμε την τιμή κ .

ένωση(α, β) : εκτελούμε $\kappa \leftarrow \text{εύρεση}(\alpha)$ και $\lambda \leftarrow \text{εύρεση}(\beta)$ · θέτουμε $\Pi(\lambda) = \kappa$

Ιδιότητα: Για n αντικείμενα και $m \geq n$ πράξεις ένωσης, ο αλγόριθμος γρήγορης ένωσης μπορεί να χρειαστεί περισσότερες από $m \cdot n / 2$ εντολές.

Απόδειξη: Η ακολουθία $\text{ένωση}(n-1, n)$, $\text{ένωση}(n-2, n)$,



Αλγόριθμοι ένωσης-εύρεσης: γρήγορη ένωση

εύρεση(α) : ξεκινώντας από το α ακολουθούμε τους δείκτες μέχρι να βρούμε βρούμε αντικείμενο κ τέτοιο ώστε $\Pi(\kappa) = \kappa$ · επιστρέφουμε την τιμή κ .

ένωση(α, β) : εκτελούμε $\kappa \leftarrow \text{εύρεση}(\alpha)$ και $\lambda \leftarrow \text{εύρεση}(\beta)$ · θέτουμε $\Pi(\lambda) = \kappa$

Ιδιότητα: Για n αντικείμενα και $m \geq n$ πράξεις ένωσης, ο αλγόριθμος γρήγορης ένωσης μπορεί να χρειαστεί περισσότερες από $m \cdot n / 2$ εντολές.

Απόδειξη: Η ακολουθία $\text{ένωση}(n-1, n), \text{ένωση}(n-2, n), \text{ένωση}(n-3, n)$



Αλγόριθμοι ένωσης-εύρεσης: γρήγορη ένωση

εύρεση(α) : ξεκινώντας από το α ακολουθούμε τους δείκτες μέχρι να βρούμε βρούμε αντικείμενο κ τέτοιο ώστε $\Pi(\kappa) = \kappa$ · επιστρέφουμε την τιμή κ .

ένωση(α, β) : εκτελούμε $\kappa \leftarrow \text{εύρεση}(\alpha)$ και $\lambda \leftarrow \text{εύρεση}(\beta)$ · θέτουμε $\Pi(\lambda) = \kappa$

Ιδιότητα: Για n αντικείμενα και $m \geq n$ πράξεις ένωσης, ο αλγόριθμος γρήγορης ένωσης μπορεί να χρειαστεί περισσότερες από $m \cdot n / 2$ εντολές.

Απόδειξη: Η ακολουθία $\text{ένωση}(n-1, n), \text{ένωση}(n-2, n), \text{ένωση}(n-3, n), \dots, \text{ένωση}(1, n)$



Αλγόριθμοι ένωσης-εύρεσης: γρήγορη ένωση

εύρεση(α) : ξεκινώντας από το α ακολουθούμε τους δείκτες μέχρι να βρούμε βρούμε αντικείμενο κ τέτοιο ώστε $\Pi(\kappa) = \kappa$ · επιστρέφουμε την τιμή κ .

ένωση(α, β) : εκτελούμε $\kappa \leftarrow \text{εύρεση}(\alpha)$ και $\lambda \leftarrow \text{εύρεση}(\beta)$ · θέτουμε $\Pi(\lambda) = \kappa$

Ιδιότητα: Για n αντικείμενα και $m \geq n$ πράξεις ένωσης, ο αλγόριθμος γρήγορης ένωσης μπορεί να χρειαστεί περισσότερες από $m n / 2$ εντολές.

Απόδειξη: Η ακολουθία $\text{ένωση}(n-1, n), \text{ένωση}(n-2, n), \text{ένωση}(n-3, n), \dots, \text{ένωση}(1, n)$

διατρέχει $\sum_{\kappa=1}^{\kappa=n-1} \kappa = \frac{n(n-1)}{2} = \mu' n / 2$ δείκτες (από το n προς τη ρίζα).



Αλγόριθμοι ένωσης-εύρεσης: γρήγορη ένωση

εύρεση(α) : ξεκινώντας από το α ακολουθούμε τους δείκτες μέχρι να βρούμε βρούμε αντικείμενο κ τέτοιο ώστε $\Pi(\kappa) = \kappa$ · επιστρέφουμε την τιμή κ .

ένωση(α, β) : εκτελούμε $\kappa \leftarrow \text{εύρεση}(\alpha)$ και $\lambda \leftarrow \text{εύρεση}(\beta)$ · θέτουμε $\Pi(\lambda) = \kappa$

Ιδιότητα: Για n αντικείμενα και $m \geq n$ πράξεις ένωσης, ο αλγόριθμος γρήγορης ένωσης μπορεί να χρειαστεί περισσότερες από $m/2$ εντολές.

Απόδειξη: Η ακολουθία $\text{ένωση}(n-1, n), \text{ένωση}(n-2, n), \text{ένωση}(n-3, n), \dots, \text{ένωση}(1, n)$

1 διατρέχει $\sum_{\kappa=1}^{\kappa=n-1} \kappa = \frac{n(n-1)}{2} = \mu' n/2$ δείκτες (από το n προς τη ρίζα).

5 Για μ'' ακόμα πράξεις του τύπου $\text{ένωση}(n, \kappa)$, έχουμε να διατρέξουμε
6 $(n-1)\mu''$ δείκτες τουλάχιστον.

7
8 Συνολικά έχουμε $n\mu'/2 + (n-1)\mu'' \geq n(\mu' + \mu'')/2 = n\mu/2$

1^η βελτίωση: σταθμισμένη γρήγορη ένωση

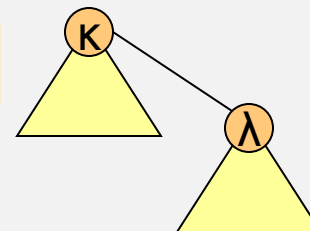
εύρεση(α) : ξεκινώντας από το α ακολουθούμε τους δείκτες μέχρι να βρούμε βρούμε αντικείμενο κ τέτοιο ώστε $\Pi(\kappa) = \kappa$ · επιστρέφουμε την τιμή κ .

Διατηρούμε μία ακόμα πληροφορία:

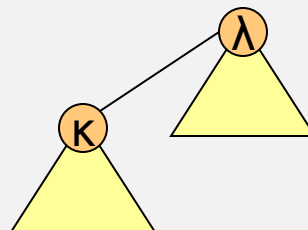
μέγεθος(α) = πλήθος αντικειμένων στο δέντρο με ρίζα α

ένωση(α, β) : εκτελούμε $\kappa \leftarrow \text{εύρεση}(\alpha)$ και $\lambda \leftarrow \text{εύρεση}(\beta)$ ·

Αν $\text{μέγεθος}(\kappa) \geq \text{μέγεθος}(\lambda)$, θέτουμε $\Pi(\lambda) = \kappa$



Διαφορετικά, θέτουμε $\Pi(\kappa) = \lambda$



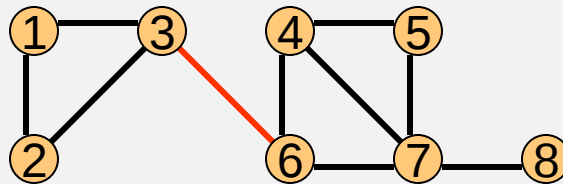
1^η βελτίωση: σταθμισμένη γρήγορη ένωση

ένωση(α, β) : εκτελούμε $\kappa \leftarrow \text{εύρεση}(\alpha)$ και $\lambda \leftarrow \text{εύρεση}(\beta)$.

Αν $\text{μέγεθος}(\kappa) \geq \text{μέγεθος}(\lambda)$, θέτουμε $\Pi(\lambda) = \kappa$

Διαφορετικά, θέτουμε $\Pi(\kappa) = \lambda$

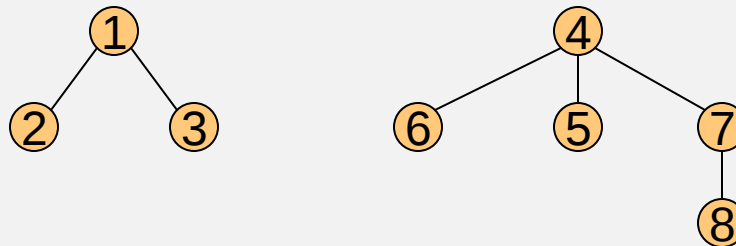
γράφημα



πίνακας

$$\Pi = [1 \quad 1 \quad 1 \quad 4 \quad 4 \quad 4 \quad 4 \quad 7]$$

δέντρο ένωσης-εύρεσης



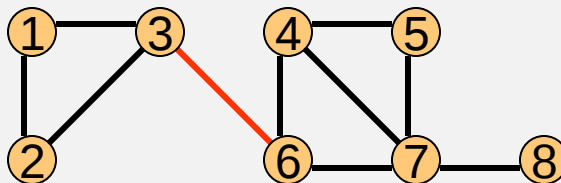
1^η βελτίωση: σταθμισμένη γρήγορη ένωση

ένωση(α, β) : εκτελούμε $\kappa \leftarrow \text{εύρεση}(\alpha)$ και $\lambda \leftarrow \text{εύρεση}(\beta)$.

Αν $\text{μέγεθος}(\kappa) \geq \text{μέγεθος}(\lambda)$, θέτουμε $\Pi(\lambda) = \kappa$

Διαφορετικά, θέτουμε $\Pi(\kappa) = \lambda$

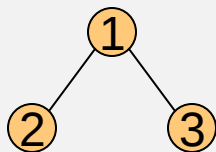
γράφημα



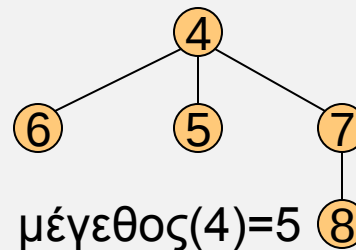
πίνακας

$$\Pi = [1 \quad 1 \quad 1 \quad 4 \quad 4 \quad 4 \quad 4 \quad 7]$$

δέντρο ένωσης-εύρεσης



μέγεθος(1)=3



μέγεθος(4)=5

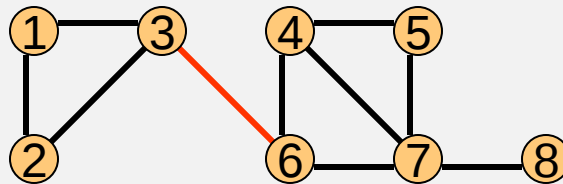
1^η βελτίωση: σταθμισμένη γρήγορη ένωση

ένωση(α, β) : εκτελούμε $\kappa \leftarrow \text{εύρεση}(\alpha)$ και $\lambda \leftarrow \text{εύρεση}(\beta)$.

Αν $\text{μέγεθος}(\kappa) \geq \text{μέγεθος}(\lambda)$, θέτουμε $\Pi(\lambda) = \kappa$

Διαφορετικά, θέτουμε $\Pi(\kappa) = \lambda$

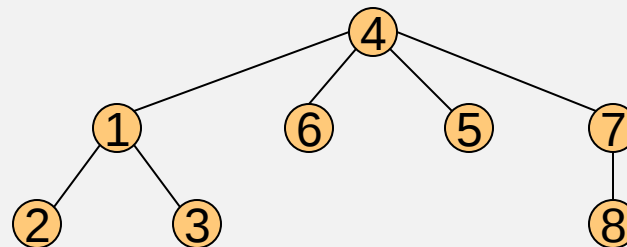
γράφημα



πίνακας

$$\Pi = [4 \quad 1 \quad 1 \quad 4 \quad 4 \quad 4 \quad 4 \quad 7]$$

δέντρο ένωσης-εύρεσης



1^η βελτίωση: σταθμισμένη γρήγορη ένωση

ένωση(α, β) : εκτελούμε $\kappa \leftarrow \text{εύρεση}(\alpha)$ και $\lambda \leftarrow \text{εύρεση}(\beta)$.

Αν $\text{μέγεθος}(\kappa) \geq \text{μέγεθος}(\lambda)$, θέτουμε $\Pi(\lambda) = \kappa$

Διαφορετικά, θέτουμε $\Pi(\kappa) = \lambda$

Ας θεωρήσουμε την ακολουθία

ένωση(1,2), ένωση(3,4), ένωση(5,6), ένωση(7,8)



1^η βελτίωση: σταθμισμένη γρήγορη ένωση

ένωση(α, β) : εκτελούμε $\kappa \leftarrow \text{εύρεση}(\alpha)$ και $\lambda \leftarrow \text{εύρεση}(\beta)$.

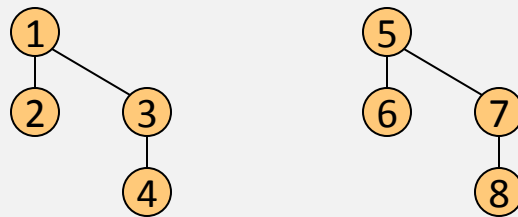
Αν $\text{μέγεθος}(\kappa) \geq \text{μέγεθος}(\lambda)$, θέτουμε $\Pi(\lambda) = \kappa$

Διαφορετικά, θέτουμε $\Pi(\kappa) = \lambda$

Ας θεωρήσουμε την ακολουθία

ένωση(1,2), ένωση(3,4), ένωση(5,6), ένωση(7,8)

ένωση(1,3), ένωση(5,7)



1^η βελτίωση: σταθμισμένη γρήγορη ένωση

ένωση(α, β) : εκτελούμε $\kappa \leftarrow \text{εύρεση}(\alpha)$ και $\lambda \leftarrow \text{εύρεση}(\beta)$.

Αν $\text{μέγεθος}(\kappa) \geq \text{μέγεθος}(\lambda)$, θέτουμε $\Pi(\lambda) = \kappa$

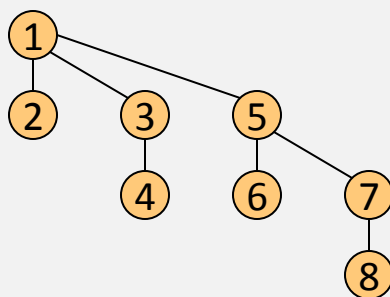
Διαφορετικά, θέτουμε $\Pi(\kappa) = \lambda$

Ας θεωρήσουμε την ακολουθία

ένωση(1,2), ένωση(3,4), ένωση(5,6), ένωση(7,8)

ένωση(1,3), ένωση(5,7)

ένωση(1,5)



1^η βελτίωση: σταθμισμένη γρήγορη ένωση

ένωση(α, β) : εκτελούμε $\kappa \leftarrow \text{εύρεση}(\alpha)$ και $\lambda \leftarrow \text{εύρεση}(\beta)$.

Αν $\text{μέγεθος}(\kappa) \geq \text{μέγεθος}(\lambda)$, θέτουμε $\Pi(\lambda) = \kappa$

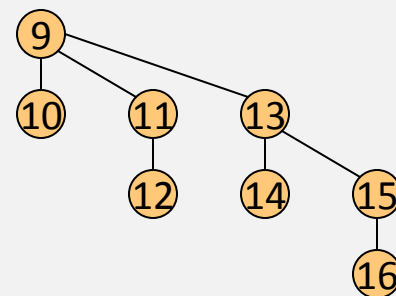
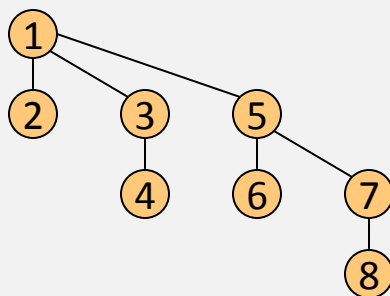
Διαφορετικά, θέτουμε $\Pi(\kappa) = \lambda$

Ομοίως, για

ένωση(9,10), ένωση(11,12), ένωση(13,14), ένωση(15,16)

ένωση(9,11), ένωση(13,15)

ένωση(9,13)



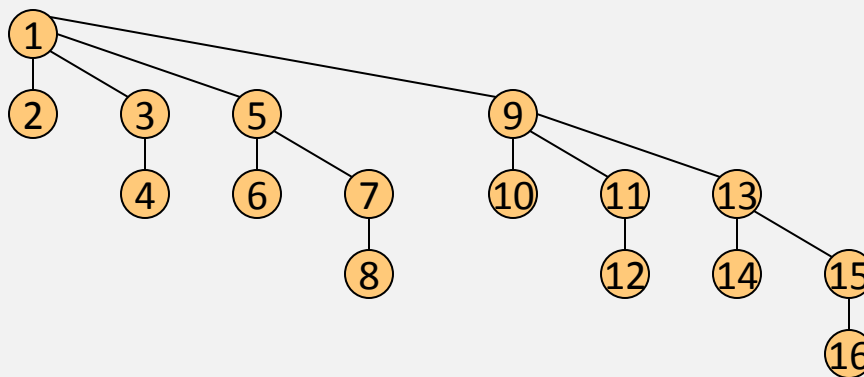
1^η βελτίωση: σταθμισμένη γρήγορη ένωση

ένωση(α, β) : εκτελούμε $\kappa \leftarrow \text{εύρεση}(\alpha)$ και $\lambda \leftarrow \text{εύρεση}(\beta)$.

Αν $\text{μέγεθος}(\kappa) \geq \text{μέγεθος}(\lambda)$, θέτουμε $\Pi(\lambda) = \kappa$

Διαφορετικά, θέτουμε $\Pi(\kappa) = \lambda$

ένωση(1,9)



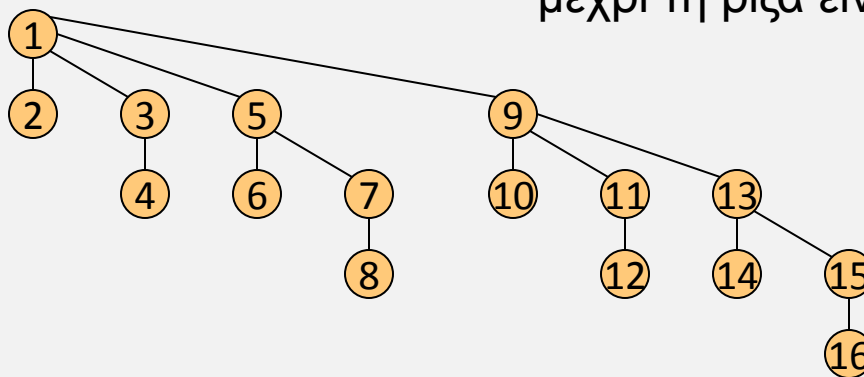
1^η βελτίωση: σταθμισμένη γρήγορη ένωση

ένωση(α, β) : εκτελούμε $\kappa \leftarrow \text{εύρεση}(\alpha)$ και $\lambda \leftarrow \text{εύρεση}(\beta)$.

Αν $\text{μέγεθος}(\kappa) \geq \text{μέγεθος}(\lambda)$, θέτουμε $\Pi(\lambda) = \kappa$

Διαφορετικά, θέτουμε $\Pi(\kappa) = \lambda$

ο μέγιστος αριθμός των δεικτών
μέχρι τη ρίζα είναι $\lg v$



1^η βελτίωση: σταθμισμένη γρήγορη ένωση

ένωση(α, β) : εκτελούμε $\kappa \leftarrow \text{εύρεση}(\alpha)$ και $\lambda \leftarrow \text{εύρεση}(\beta)$.

Αν $\text{μέγεθος}(\kappa) \geq \text{μέγεθος}(\lambda)$, θέτουμε $\Pi(\lambda) = \kappa$

Διαφορετικά, θέτουμε $\Pi(\kappa) = \lambda$

Ιδιότητα: Για n αντικείμενα, ο αλγόριθμος σταθμισμένης γρήγορης ένωσης δημιουργεί δέντρα με ύψος το πολύ $\lg n$

1^η βελτίωση: σταθμισμένη γρήγορη ένωση

ένωση(α, β) : εκτελούμε $\kappa \leftarrow \text{εύρεση}(\alpha)$ και $\lambda \leftarrow \text{εύρεση}(\beta)$.

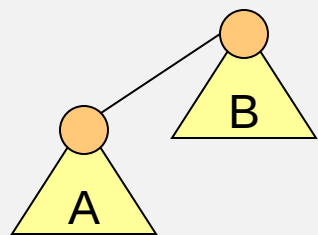
Αν $\text{μέγεθος}(\kappa) \geq \text{μέγεθος}(\lambda)$, θέτουμε $\Pi(\lambda) = \kappa$

Διαφορετικά, θέτουμε $\Pi(\kappa) = \lambda$

Ιδιότητα: Για n αντικείμενα, ο αλγόριθμος σταθμισμένης γρήγορης ένωσης δημιουργεί δέντρα με ύψος το πολύ $\lg n$

Απόδειξη: Με επαγωγή. Για $n=2$ είναι προφανές ότι ισχύει.

Έστω ότι μια πράξη ένωσης συνδυάζει σύνολα A και B με μέγεθος $|A| = \alpha \leq |B| = \beta$



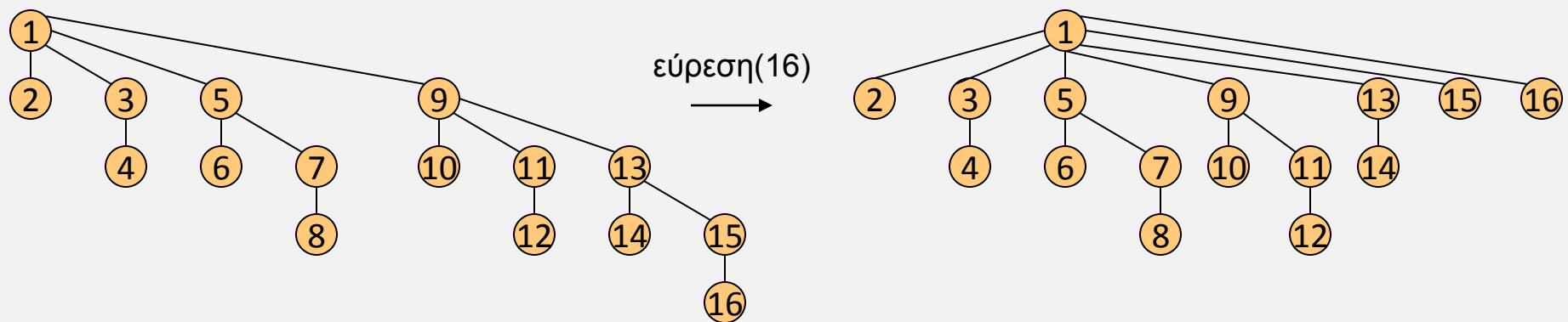
Ο αριθμός των δεικτών έως τη ρίζα του δέντρου για τα στοιχεία του A είναι $\lg \alpha + 1 = \lg \alpha + \lg 2 = \lg(2\alpha) \leq \lg(\alpha + \beta)$

2^η βελτίωση: συμπίεση διαδρομής

εύρεση(α) : ξεκινώντας από το α ακολουθούμε τους δείκτες μέχρι να βρούμε βρούμε τη ρίζα κ του δέντρου. Για κάθε αντικείμενο χ που συναντούμε, θέτουμε $\Pi(\chi) = \kappa$. Τέλος επιστρέφουμε την τιμή κ .

2^η βελτίωση: συμπίεση διαδρομής

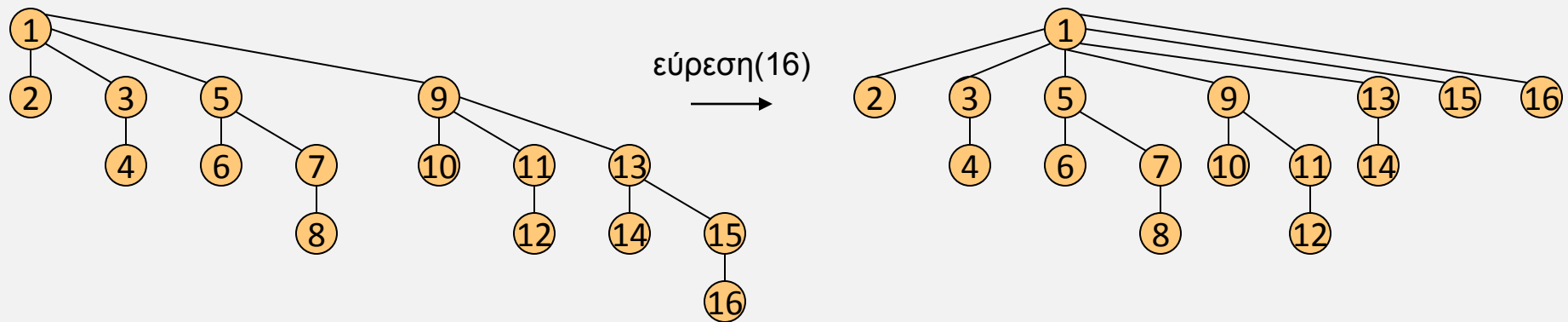
εύρεση(α) : ξεκινώντας από το α ακολουθούμε τους δείκτες μέχρι να βρούμε βρούμε τη ρίζα κ του δέντρου. Για κάθε αντικείμενο χ που συναντούμε, θέτουμε $\Pi(\chi) = \kappa$. Τέλος επιστρέφουμε την τιμή κ .



Ο χρόνος εκτέλεσης για οποιαδήποτε ακολουθία $\mu \geq n$ πράξεων εύρεσης-ένωσης με n αντικείμενα είναι σχεδόν γραμμική συνάρτηση των μ , n .

2^η βελτίωση: συμπίεση διαδρομής

εύρεση(α) : ξεκινώντας από το α ακολουθούμε τους δείκτες μέχρι να βρούμε
βρούμε τη ρίζα κ του δέντρου. Για κάθε αντικείμενο χ που συναντούμε,
θέτουμε $\Pi(\chi) = \kappa$. Τέλος επιστρέφουμε την τιμή κ .



Ο χρόνος εκτέλεσης για οποιαδήποτε ακολουθία $\mu \geq n$ πράξεων εύρεσης-ένωσης με n αντικείμενα είναι σχεδόν γραμμική συνάρτηση των μ, n .

Χρόνος εκτέλεσης : $O(\mu \alpha(\mu, n))$, όπου $\alpha(\mu, n)$ η αντίστροφη συνάρτηση Ackerman

Συνάρτηση Ackerman

Η συνάρτηση Ackerman $A(i, j)$ ορίζεται ως εξής

$$A(1, j) = 2^j, j \geq 1$$

$$A(i, 1) = A(i - 1, 2), i \geq 2$$

$$A(i, j) = A(i - 1, A(i, j - 1)), i, j \geq 2$$

Η συνάρτηση $\alpha(\mu, \nu)$ είναι αντίστροφη της $A(i, j)$

⇒ αυξάνει με πάρα πολύ αργό ρυθμό!

Π.χ. $\alpha(\mu, \nu) \leq 3$ για $\nu < 2^{16} = 65536$

2^η βελτίωση: συμπίεση διαδρομής

εύρεση(α) : ξεκινώντας από το α ακολουθούμε τους δείκτες μέχρι να βρούμε βρούμε τη ρίζα κ του δέντρου. Για κάθε αντικείμενο χ που συναντούμε, θέτουμε $\Pi(\chi) = \kappa$. Τέλος επιστρέφουμε την τιμή κ.

Ο χρόνος εκτέλεσης για οποιαδήποτε ακολουθία $\mu \geq \nu$ πράξεων εύρεσης-ένωσης με ν αντικείμενα είναι σχεδόν γραμμική συνάρτηση των μ, ν .

Χρόνος εκτέλεσης : $O(\mu\alpha(\mu, \nu))$, όπου $\alpha(\mu, \nu)$ η αντίστροφη συνάρτηση Ackerman

- ⇒ Μια εντολή εύρεσης ή ένωση μπορεί να πάρει $\Theta(\lg \nu)$ χρόνο **στη χειρότερη περίπτωση**, ωστόσο...
- ⇒ ακόμα και στη **χειρότερη δυνατή ακολουθία** μια λειτουργία εύρεσης ή ένωσης παίρνει $O(\alpha(\mu, \nu))$ χρόνο **κατά μέσο όρο**

2^η βελτίωση: συμπίεση διαδρομής

εύρεση(α) : ξεκινώντας από το α ακολουθούμε τους δείκτες μέχρι να βρούμε βρούμε τη ρίζα κ του δέντρου. Για κάθε αντικείμενο χ που συναντούμε, θέτουμε $\Pi(\chi) = \kappa$. Τέλος επιστρέφουμε την τιμή κ.

Ο χρόνος εκτέλεσης για οποιαδήποτε ακολουθία $\mu \geq \nu$ πράξεων εύρεσης-ένωσης με ν αντικείμενα είναι σχεδόν γραμμική συνάρτηση των μ, ν .

Χρόνος εκτέλεσης : $O(\mu\alpha(\mu, \nu))$, όπου $\alpha(\mu, \nu)$ η αντίστροφη συνάρτηση Ackerman

- ⇒ Μια εντολή εύρεσης ή ένωση μπορεί να πάρει $\Theta(\lg \nu)$ χρόνο **στη χειρότερη περίπτωση**, ωστόσο...
- ⇒ ακόμα και στη **χειρότερη δυνατή ακολουθία** μια λειτουργία εύρεσης ή ένωσης παίρνει $O(\alpha(\mu, \nu))$ χρόνο **κατά μέσο όρο**

Αντισταθμιστική Ανάλυση