

25/5/2016

ΟΥΡΕΣ ΠΡΟΤΕΡΑΙΟΤΗΤΑΣ

Δομές Δεδομένων

Τι θα δούμε

- ⊙ **Ουρές προτεραιότητας – Πράξεις**
- ⊙ **Διωνυμικές Ουρές**
 - ⊙ Διωνυμικά Δέντρα
 - ⊙ Διωνυμικοί Σωροί
- ⊙ **Ουρές Fibonacci**
 - ⊙ Αναπαράσταση
 - ⊙ Πράξεις
 - ⊙ Ανάλυση
- ⊙ **Συγκρίσεις**

Ουρές προτεραιότητας – Priority Queue

Συχνά χρειαζόμαστε δομές δεδομένων για την εισαγωγή στοιχείων που:

- το καθένα έχει κάποια **προτεραιότητα**
- η σειρά διαγραφής να καθορίζεται από **προτεραιότητα** (**μεγαλύτερη** - μικρότερη), χωρίς να μας ενδιαφέρει η σειρά με την οποία έγινε η εισαγωγή

Ουρές προτεραιότητας – Priority Queue

Συχνά χρειαζόμαστε δομές δεδομένων για την εισαγωγή στοιχείων που:

- το καθένα έχει κάποια **προτεραιότητα**
- η σειρά διαγραφής να καθορίζεται από **προτεραιότητα** (**μεγαλύτερη** - μικρότερη), χωρίς να μας ενδιαφέρει η σειρά με την οποία έγινε η εισαγωγή

Μια τέτοια ουρά ονομάζεται **ουρά προτεραιότητας**

Ορισμός

Τι είναι η Ουρά Προτεραιότητας;

Ορισμός

Τι είναι η Ουρά Προτεραιότητας;

Είναι ένας Αφηρημένος Τύπος Δεδομένων που:

- Διατηρεί ένα **σύνολο στοιχείων S**, όπου κάθε στοιχείο έχει μια **συσχετισμένη τιμή key(v)**, η οποία υποδηλώνει την προτεραιότητα του στοιχείου.

Εφαρμογές

Τι είναι η Ουρά Προτεραιότητας;

Είναι ένας Αφηρημένος Τύπος Δεδομένων που:

- Διατηρεί ένα **σύνολο στοιχείων S**, όπου κάθε στοιχείο έχει μια **συσχετισμένη τιμή key(v)**, η οποία υποδηλώνει την προτεραιότητα του στοιχείου.
- Υποστηρίζει τις πράξεις:
 - **MakeQueue()**: Δημιουργία κενής ουράς
 - **Insert(Q,x)**: Εισαγωγή του στοιχείου x στην ουρά Q
 - **Delete(Q,x)**: Διαγραφή του στοιχείου x από την ουρά Q
 - **FindMin(Q)**: Επιστρέφει ένα δείκτη στην ελάχιστη τιμή της Q
 - **DeleteMin(Q)**: Διαγραφή του στοιχείου που περιέχει την μικρότερη τιμή
 - **Meld(Q₁, Q₂)**: Ένωση των ουρών Q₁ και Q₂
 - **DecreaseKey(Q,x,k)**: Ανάθεση της τιμής k στο κλειδί που είναι αποθηκευμένο στο στοιχείο x της ουράς Q.

Εφαρμογές

- ◉ **Άμεσες εφαρμογές:**
 - ◉ Υλοποίηση ουρών αναμονής με προτεραιότητες
 - ◉ Δρομολόγηση με προτεραιότητες
 - ◉ Largest (Smallest) Processing Time First

- ◉ **Έμμεσες εφαρμογές:**
 - ◉ Βασικό συστατικό πολλών $\Delta\Delta$ και αλγορίθμων:
 - ◉ HeapSort
 - ◉ Αλγόριθμος Huffman
 - ◉ Αλγόριθμος Prim

Πιθανές υλοποιήσεις

1. Συνδεδεμένη λίστα
2. Ταξινομημένη συνδεδεμένη λίστα
3. Δυαδικό δένδρο αναζήτησης

Πιθανές υλοποιήσεις

1. Συνδεδεμένη λίστα
2. Ταξινομημένη συνδεδεμένη λίστα
3. Δυαδικό δένδρο αναζήτησης

Υπάρχει καλύτερη υλοποίηση;

Πιθανές υλοποιήσεις

1. Συνδεδεμένη λίστα
2. Ταξινομημένη συνδεδεμένη λίστα
3. Δυαδικό δένδρο αναζήτησης

Υπάρχει καλύτερη υλοποίηση;

Οι ουρές προτεραιότητας είναι μία παραλλαγή της κλασσικής σωρού

Τι είναι η Διωνυμική Ουρά Προτεραιότητας;

Είναι ένα δάσος που αποτελείται από ένα αριθμό δένδρων, τα οποία ονομάζονται **Διωνυμικά Δένδρα (Binomial Trees)**

Διωνυμικό Δέντρο

Συμβολίζω ένα Διωνυμικό Δέντρο έστω B_k

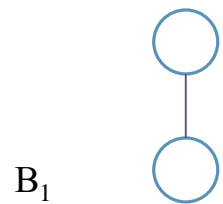
Ορισμός

- i. Το δέντρο B_0 αποτελείται από ένα μόνο στοιχείο
- ii. Το δέντρο B_k , $k \geq 1$, αποτελείται από 2 B_{k-1} δέντρα που συνδέονται μεταξύ τους έτσι ώστε η ρίζα του ενός να είναι το αριστερότερο παιδί της ρίζας του άλλου

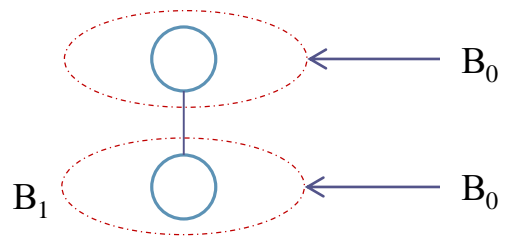
Διωνυμικό Δέντρο

B_0 

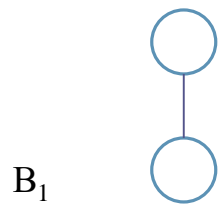
Διωνυμικό Δέντρο



Διωνυμικό Δέντρο

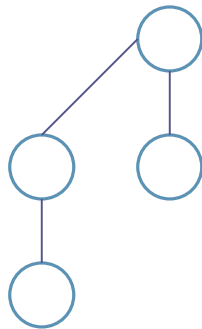


Διωνυμικό Δέντρο



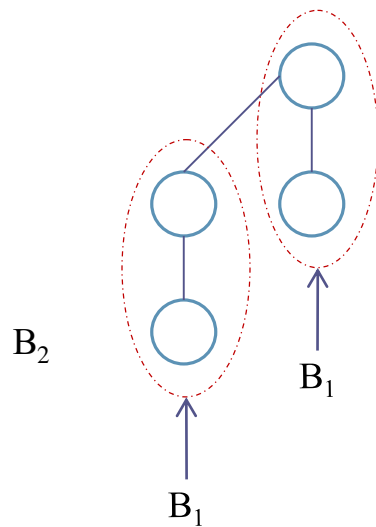
Διωνυμικό Δέντρο

B_2



Ουρές Προτεραιότητας

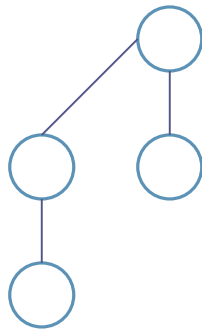
Διωνυμικό Δέντρο



Ουρές Προτεραιότητας

Διωνυμικό Δέντρο

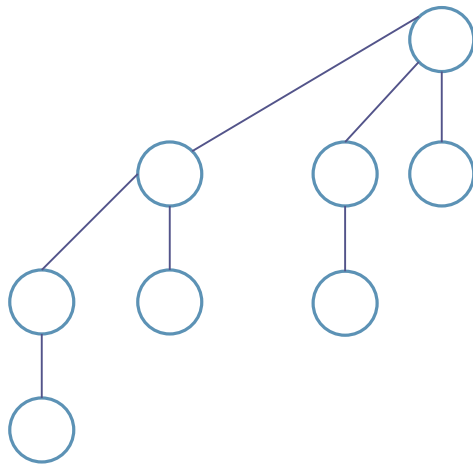
B_2



Ουρές Προτεραιότητας

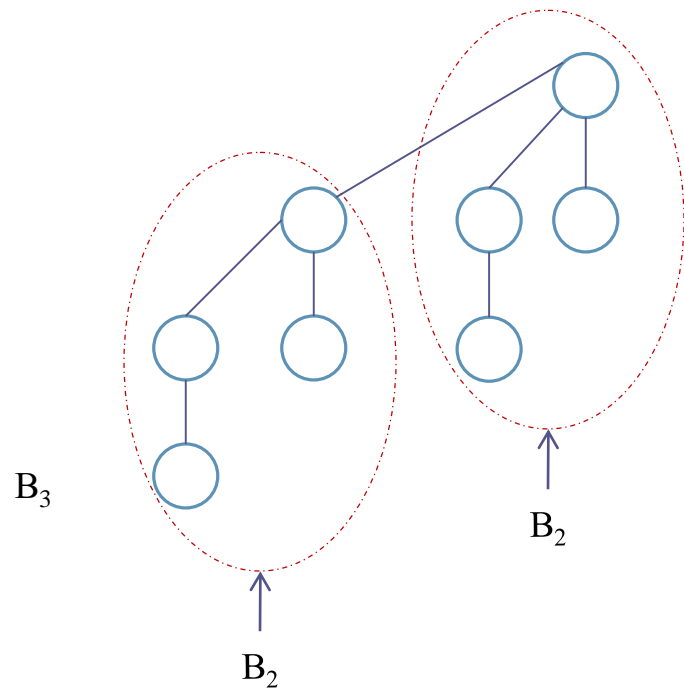
Διωνυμικό Δέντρο

B_3



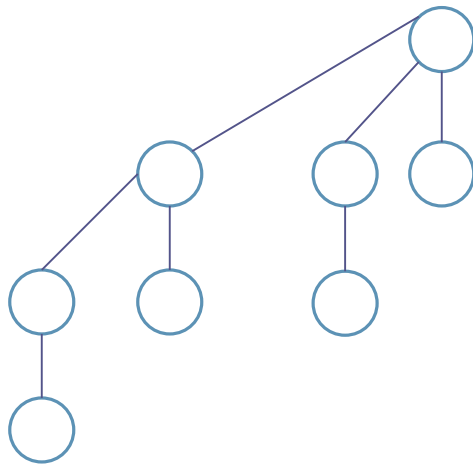
Ουρές Προτεραιότητας

Διωνυμικό Δέντρο



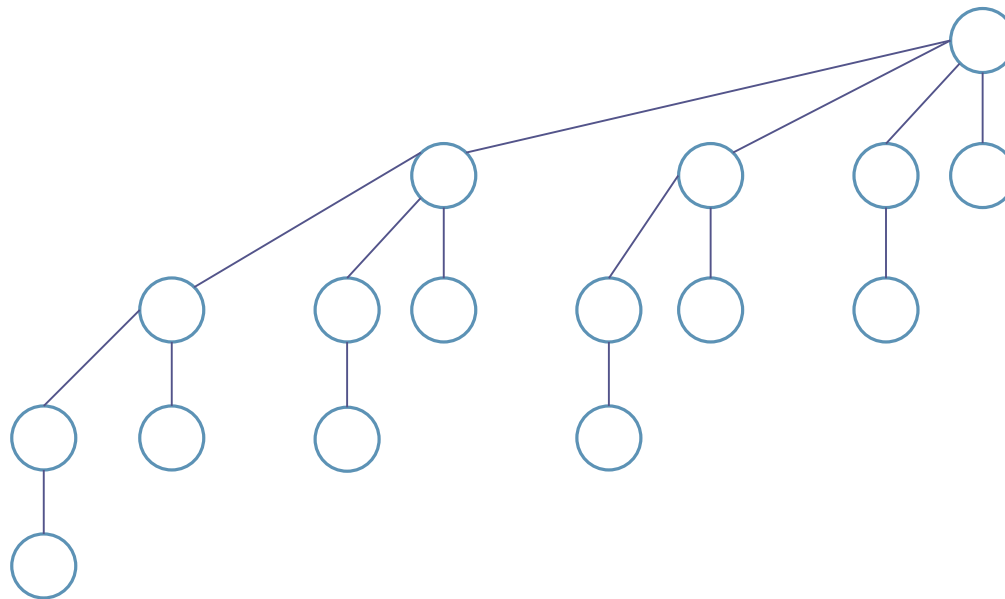
Διωνυμικό Δέντρο

B_3



Ουρές Προτεραιότητας

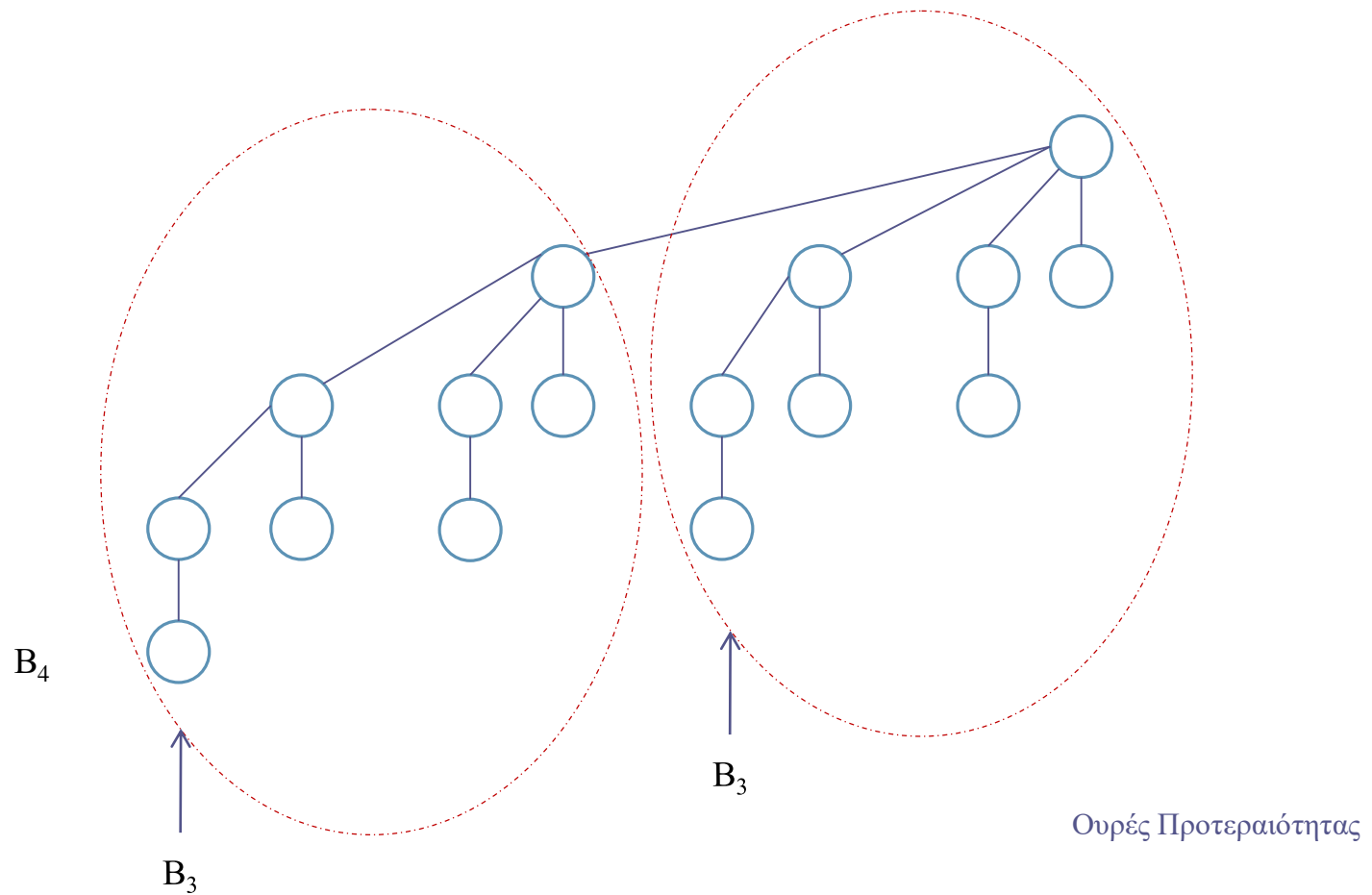
Διωνυμικό Δέντρο



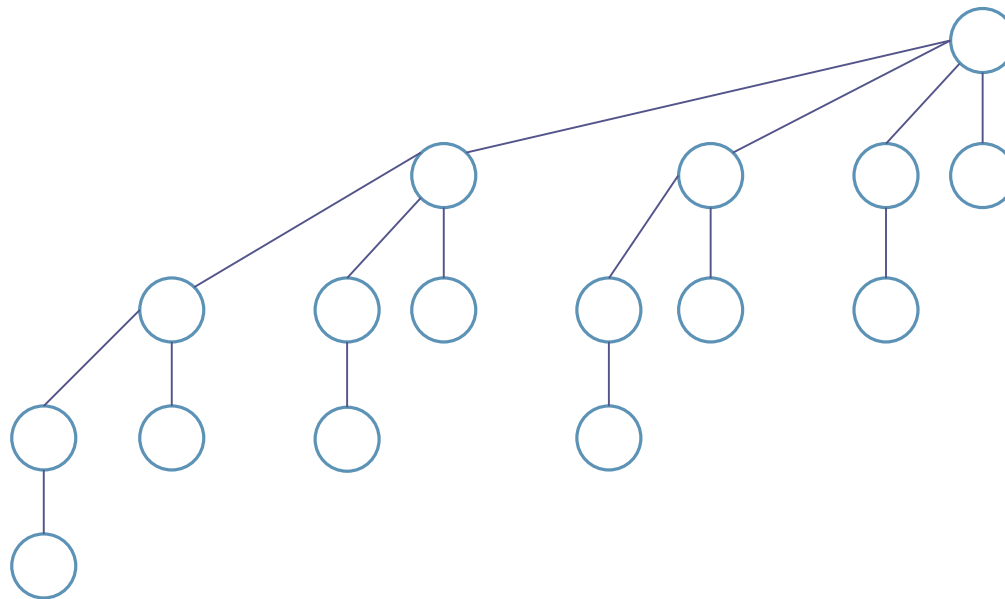
B₄

Ουρές Προτεραιότητας

Διωνυμικό Δέντρο



Διωνυμικό Δέντρο



B_4

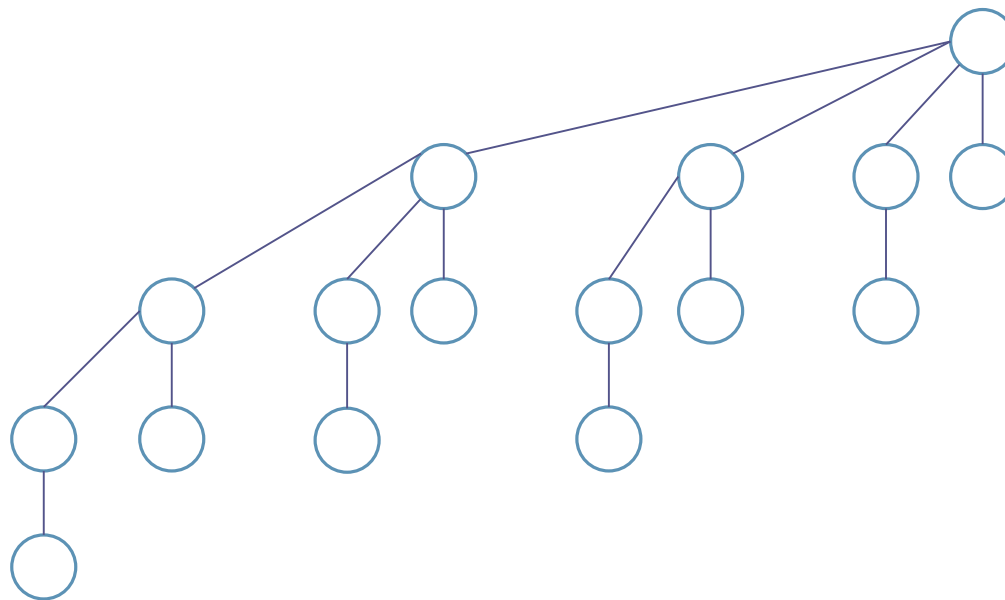
Ουρές Προτεραιότητας

Διωνυμικό Δέντρο

Τι παρατηρούμε;

1. Το δέντρο \mathbf{B}_k έχει 2^k κόμβους
2. Το δέντρο έχει ύψος k
3. Στο επίπεδο i υπάρχουν $\binom{k}{i}$ κόμβοι
4. Η ρίζα του δέντρου \mathbf{B}_k έχει k παιδιά

Διωνυμικό Δέντρο

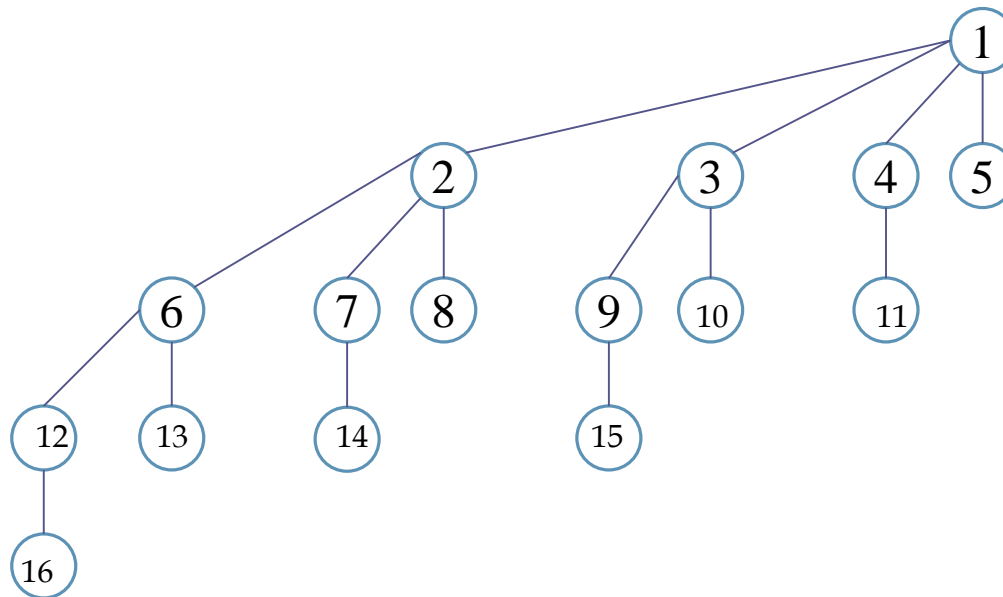


B_4

Ουρές Προτεραιότητας

Διωνυμικό Δέντρο

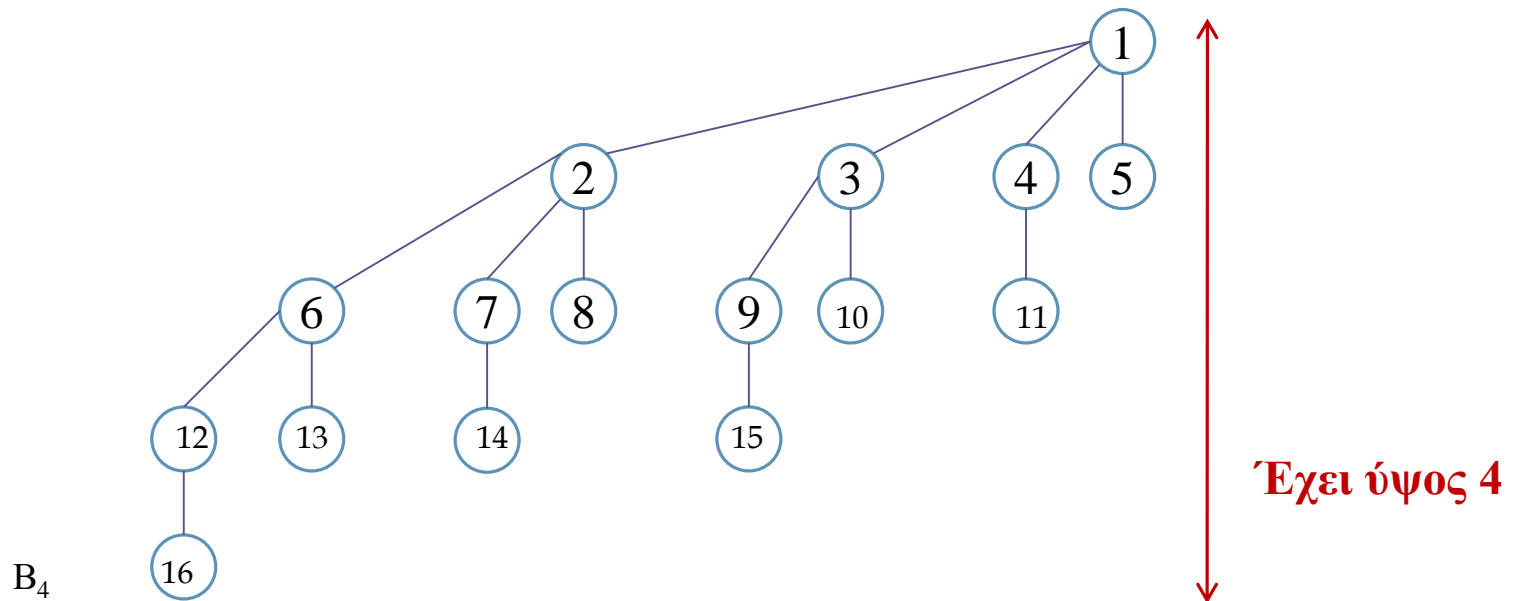
Το B_4 έχει 16 κόμβους



B_4

Διωνυμικό Δέντρο

Το B_4 έχει 16 κόμβους



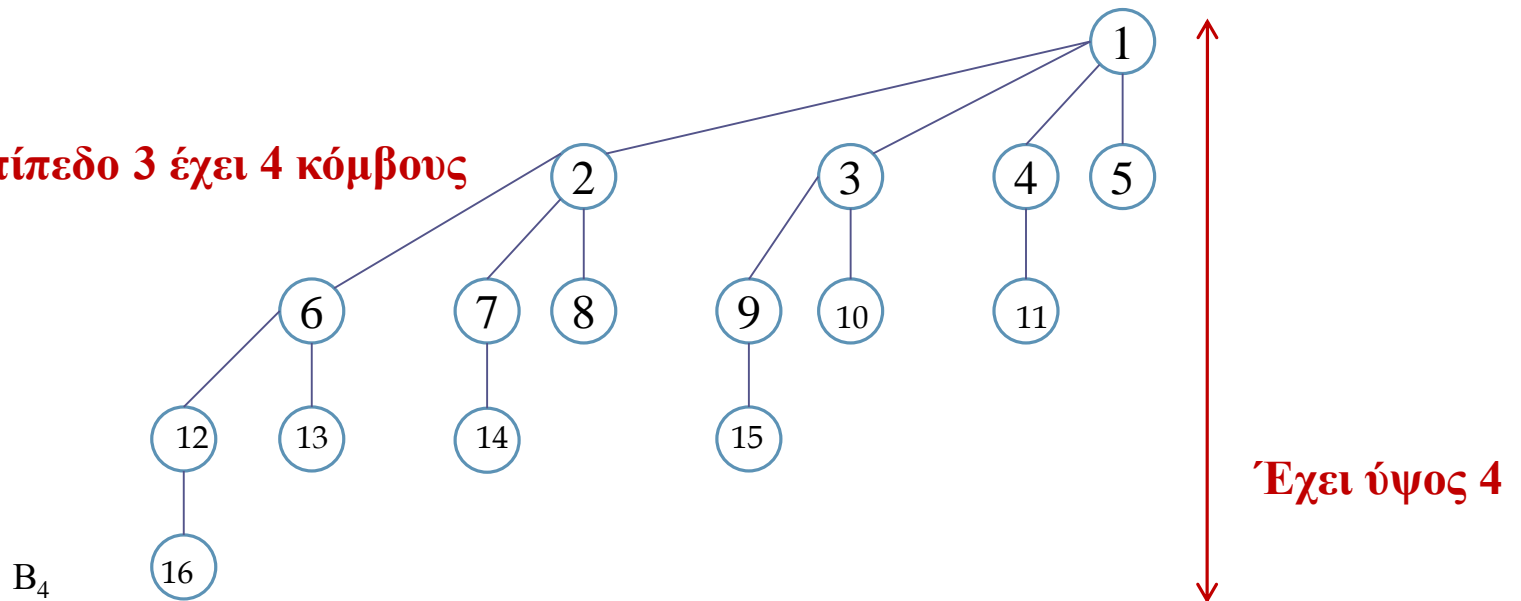
B_4

Ουρές Προτεραιότητας

Διωνυμικό Δέντρο

Το B_4 έχει 16 κόμβους

Στο επίπεδο 3 έχει 4 κόμβους



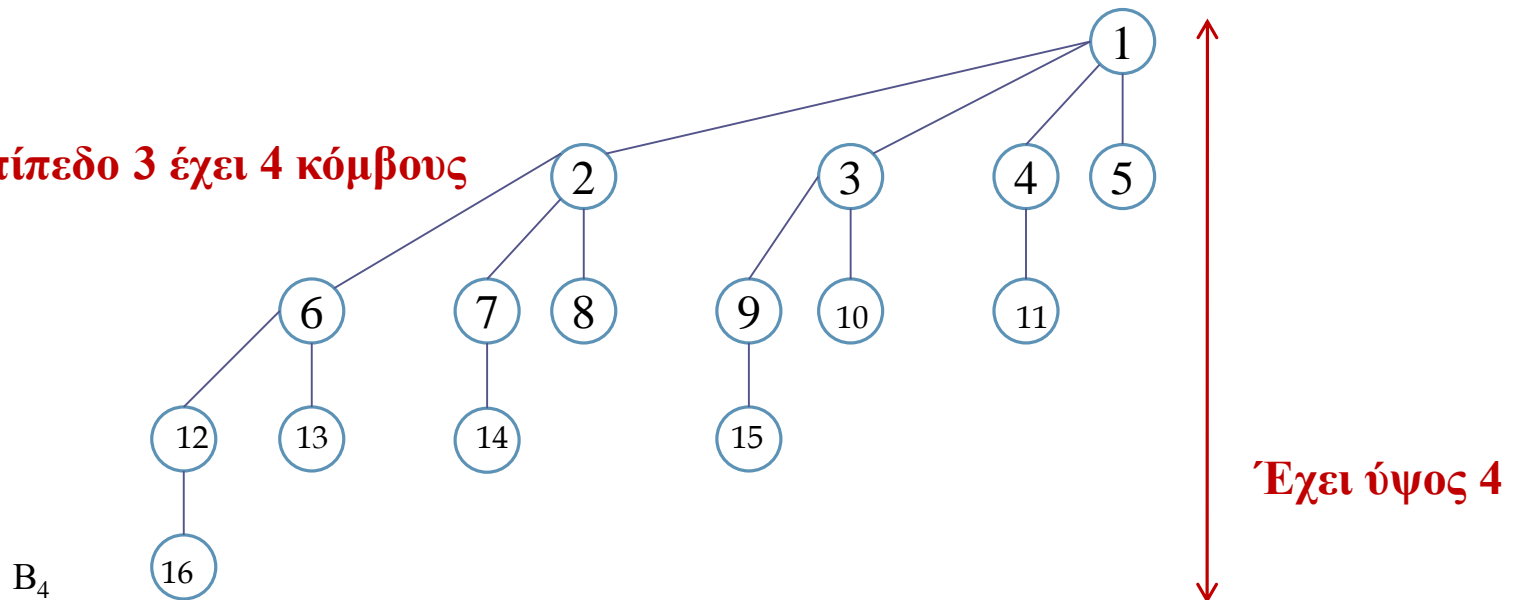
Ουρές Προτεραιότητας

Διωνυμικό Δέντρο

Το B_4 έχει 16 κόμβους

Η ρίζα έχει 4 παιδιά

Στο επίπεδο 3 έχει 4 κόμβους



Ουρές Προτεραιότητας

Τι είναι ο Διωνυμικός Σωρός;

Ένα σύνολο Διωνυμικών Δέντρων

ή αλλιώς

Ένα δάσος Διωνυμικών Δέντρων

Διωνυμικός Σωρός

- ⊙ Οι ρίζες των δέντρων είναι οργανωμένες σε διασυνδεδεμένη λίστα → *Λίστα Ριζών*
- ⊙ Σε κάθε Διωνυμικό Δέντρο ενός Διωνυμικού Σωρού η τιμή που είναι αποθηκευμένη στο γονέα είναι **μικρότερη** από την τιμή/τις τιμές που είναι αποθηκευμένη/ες **στα παιδιά του**.

Διωνυμικός Σωρός

- ⊙ Οι ρίζες των δέντρων είναι οργανωμένες σε διασυνδεδεμένη λίστα → *Λίστα Ριζών*
- ⊙ Σε κάθε Διωνυμικό Δέντρο ενός Διωνυμικού Σωρού η τιμή που είναι αποθηκευμένη στο γονέα είναι **μικρότερη** από την τιμή/τις τιμές που είναι αποθηκευμένη/ες **στα παιδιά του**.

Που θα αποθηκευτεί η μικρότερη τιμή;;

Διωνυμικός Σωρός

- Οι ρίζες των δέντρων είναι οργανωμένες σε διασυνδεδεμένη λίστα → **Λίστα Ριζών**
- Σε κάθε Διωνυμικό Δέντρο ενός Διωνυμικού Σωρού η τιμή που είναι αποθηκευμένη στο γονέα είναι **μικρότερη** από την τιμή/τις τιμές που είναι αποθηκευμένη/ες **στα παιδιά του**.

Που θα αποθηκευτεί η μικρότερη τιμή;;

- **Ανάλογα με τον αριθμό κόμβων δημιουργούμε τον Διωνυμικό Σωρό:**
 - Μετατρέπουμε τον αριθμό σε δυαδικό
 - Με βάση την δύναμη του 2 που αντιστοιχεί σε 1 φτιάχνουμε τα αντίστοιχα Διωνυμικά Δέντρα
 - Τοποθετούμε τα Διωνυμικά Δέντρα από αριστερά προς δεξιά κατά αύξουσα τάξη

Παράδειγμα - Διωνυμικός Σωρός 7 στοιχείων

$$7_{(10)} = 0111_{(2)} \rightarrow 7_{(10)} = 2^2 + 2^1 + 2^0$$

Παράδειγμα - Διωνυμικός Σωρός 7 στοιχείων

$$7_{(10)} = 0111_{(2)} \rightarrow 7_{(10)} = 2^2 + 2^1 + 2^0$$

Άρα ο Διωνυμικός Σωρός 7 στοιχείων θα αποτελείται από:

- 1 Διωνυμικό Δέντρο \mathbf{B}_0

Παράδειγμα - Διωνυμικός Σωρός 7 στοιχείων

$$7_{(10)} = 0111_{(2)} \rightarrow 7_{(10)} = 2^2 + 2^1 + 2^0$$

Άρα ο Διωνυμικός Σωρός 7 στοιχείων θα αποτελείται από:

- 1 Διωνυμικό Δέντρο B_0



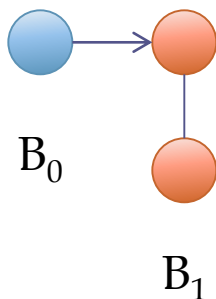
B_0

Παράδειγμα - Διωνυμικός Σωρός 7 στοιχείων

$$7_{(10)} = 0111_{(2)} \rightarrow 7_{(10)} = 2^2 + 2^1 + 2^0$$

Άρα ο Διωνυμικός Σωρός 7 στοιχείων θα αποτελείται από:

- 1 Διωνυμικό Δέντρο \mathbf{B}_0
- 1 Διωνυμικό Δέντρο \mathbf{B}_1

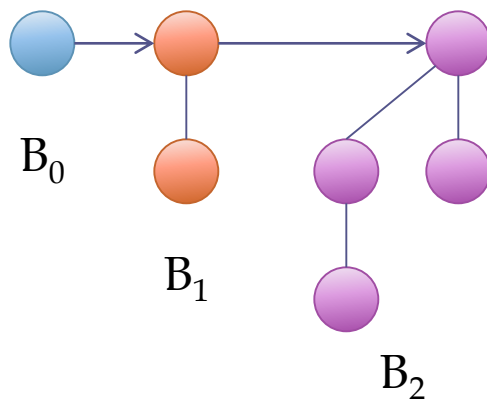


Παράδειγμα - Διωνυμικός Σωρός 7 στοιχείων

$$7_{(10)} = 0111_{(2)} \rightarrow 7_{(10)} = 2^2 + 2^1 + 2^0$$

Άρα ο Διωνυμικός Σωρός 7 στοιχείων θα αποτελείται από:

- 1 Διωνυμικό Δέντρο B_0
- 1 Διωνυμικό Δέντρο B_1
- 1 Διωνυμικό Δέντρο B_2

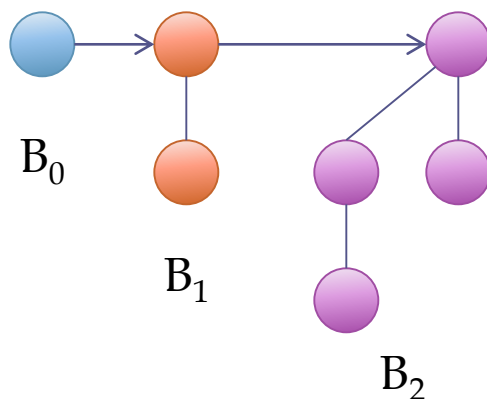


Παράδειγμα - Διωνυμικός Σωρός 7 στοιχείων

$$7_{(10)} = 0111_{(2)} \rightarrow 7_{(10)} = 2^2 + 2^1 + 2^0$$

Άρα ο Διωνυμικός Σωρός 7 στοιχείων θα αποτελείται από:

- 1 Διωνυμικό Δέντρο B_0
- 1 Διωνυμικό Δέντρο B_1
- 1 Διωνυμικό Δέντρο B_2



Παρατηρούμε ότι

ΚΑΘΕ ΡΙΖΑ ΕΧΕΙ
ΔΙΑΦΟΡΕΤΙΚΟ
ΒΑΘΜΟ!

Παράδειγμα - Διωνυμικός Σωρός 13 στοιχείων

$$13_{(10)} = 1101_{(2)} \rightarrow 13_{(10)} = 2^3 + 2^2 + 2^0$$

Παράδειγμα - Διωνυμικός Σωρός 13 στοιχείων

$$13_{(10)} = 1101_{(2)} \rightarrow 13_{(10)} = 2^3 + 2^2 + 2^0$$

Άρα ο Διωνυμικός Σωρός 13 στοιχείων θα αποτελείται από:

- 1 Διωνυμικό Δέντρο \mathbf{B}_0

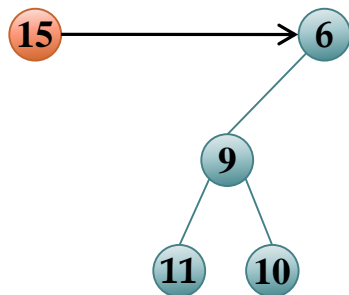
15

Παράδειγμα - Διωνυμικός Σωρός 13 στοιχείων

$$13_{(10)} = 1101_{(2)} \rightarrow 13_{(10)} = 2^3 + 2^2 + 2^0$$

Άρα ο Διωνυμικός Σωρός 13 στοιχείων θα αποτελείται από:

- 1 Διωνυμικό Δέντρο B_0
- 1 Διωνυμικό Δέντρο B_2

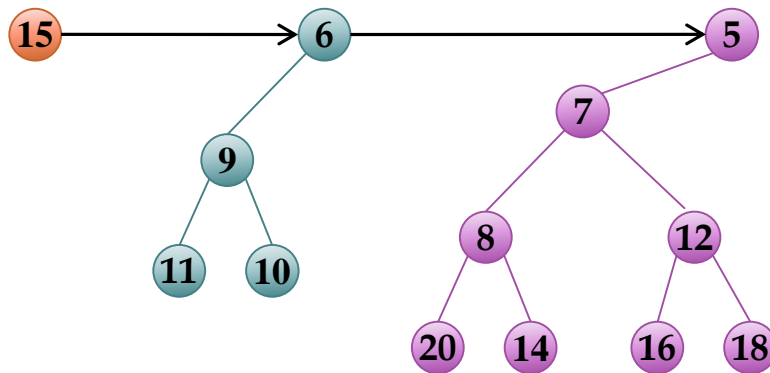


Παράδειγμα - Διωνυμικός Σωρός 13 στοιχείων

$$13_{(10)} = 1101_{(2)} \rightarrow 13_{(10)} = 2^3 + 2^2 + 2^0$$

Άρα ο Διωνυμικός Σωρός 13 στοιχείων θα αποτελείται από:

- 1 Διωνυμικό Δέντρο B_0
- 1 Διωνυμικό Δέντρο B_2
- 1 Διωνυμικό Δέντρο B_3



Οι τιμές που είναι αποθηκευμένες στους γονείς είναι μικρότερη από τις τιμές στα παιδιά

Αναπαράσταση Διωνυμικών Σωρών

Χρησιμοποιείται η τεχνική **αριστερό παιδί - δεξιός γείτονας**, δηλαδή κάθε κόμβος αποθηκεύει

- ⊙ ένα δείκτη για το αριστερό παιδί του
- ⊙ ένα δείκτη για το δεξιό γείτονά του

Και επιπλέον

- ⊙ ένα δείκτη στο γονέα του
- ⊙ ένα πεδίο με τον βαθμό του
- ⊙ ένα πεδίο με την τιμή που έχει αποθηκευμένη σε αυτόν
- ⊙ ένα πεδίο με διάφορα δεδομένα που δεν χρησιμοποιούνται στο σωρό

Πράξεις σε Διωνυμικούς Σωρούς

1. **MakeQueue()**: Δημιουργεί έναν κενό σωρό σε $\Theta(1)$ χρόνο και μας επιστρέφει ένα δείκτη σε αυτόν το σωρό.
2. **FindMin(Q)**: Για να βρούμε το ελάχιστο στοιχείο κάνουμε σειριακή αναζήτηση στη λίστα ριζών και βάζουμε ένα δείκτη στη ρίζα με την μικρότερη τιμή.
3. **Meld(Q₁, Q₂)**: Για να ενώσουμε 2 ουρές (2 διωνυμικούς σωρούς) εκτελούμε 2 στάδια
 - Συγχωνεύουμε τις λίστες ριζών, προσέχοντας κάθε λίστα των ουρών να έχουν μοναδικό βαθμό
 - Μετά τη συγχώνευση οι βαθμοί στη λίστα που προκύπτει εμφανίζονται σε αύξουσα σειρά

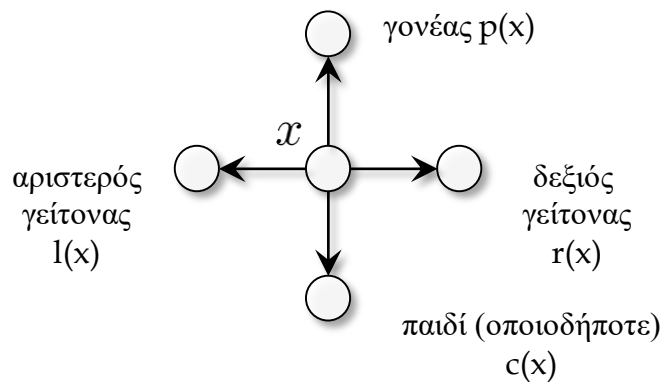
Για πιο εύκολα παραλληλίστε την παραπάνω διαδικασία με την πρόσθεση δυαδικών αριθμών

Ουρά Fibonacci

- ⦿ Βασίζεται στο Διωνυμικό Σωρό (δηλαδή αποτελεί μια συλλογή από δένδρα) αλλά έχει πιο χαλαρή δομή.
- ⦿ Καθένα από τα δέντρα ικανοποιεί την ιδιότητα του σωρού.
- ⦿ Η αναπαράσταση του δάσους γίνεται με μία διπλά συνδεδεμένη κυκλική λίστα των ριζών των δέντρων.

Αναπαράσταση Ουράς Fibonacci

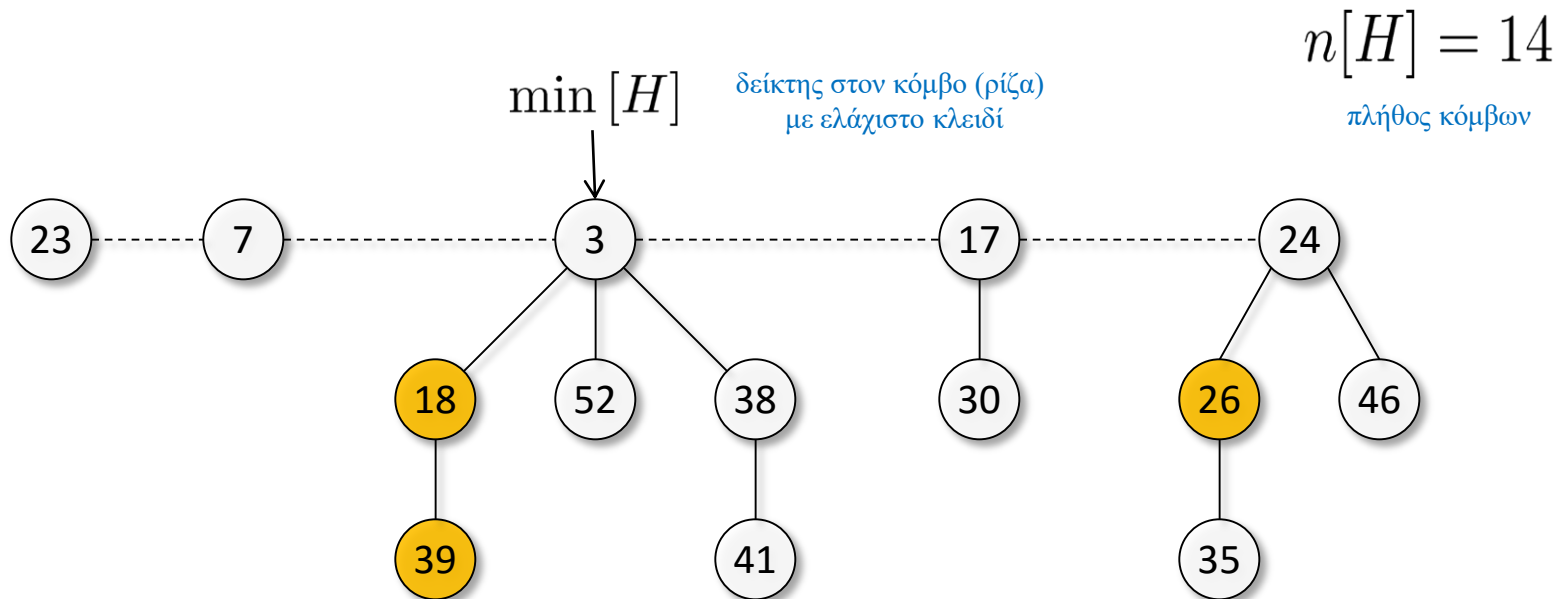
- ◉ Σε κάθε κόμβο x έχουμε:
 - ◉ $d(x)$: αριθμός παιδιών
 - ◉ $mark(x)$: boolean πεδίο, 1 όταν ο x χάνει ένα παιδί
 - ◉ $p(x)$: δείκτης προς τον γονέα
 - ◉ $c(x)$: δείκτης προς κάποιο παιδί
 - ◉ $l(x)$ και $r(x)$: δείκτες στον αριστερό και δεξιό του γείτονα αντίστοιχα



αριθμός παιδιών $d(x)$
bit επισήμανσης $mark(x)$

Ουρές Προτεραιότητας

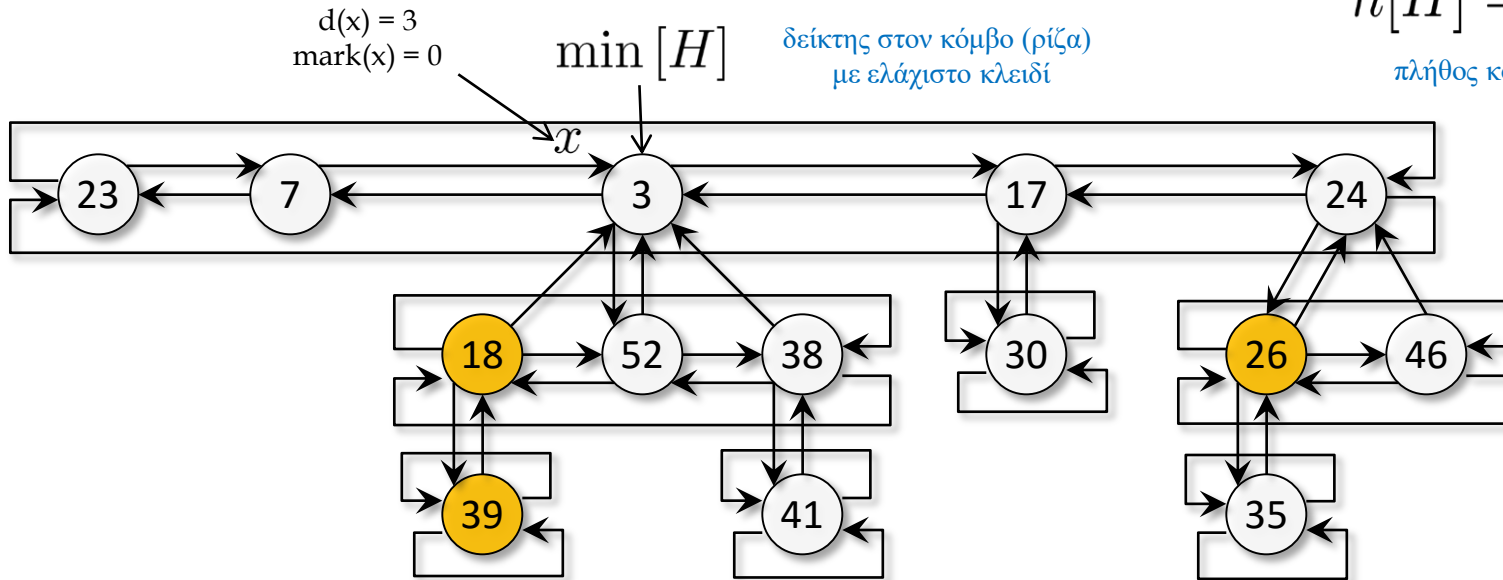
Αναπαράσταση Ουράς Fibonacci



Η ουράς Fibonacci είναι πιο χαλαρή δομή από την διωνυμική ουρά

Αναπαράσταση Ουράς Fibonacci

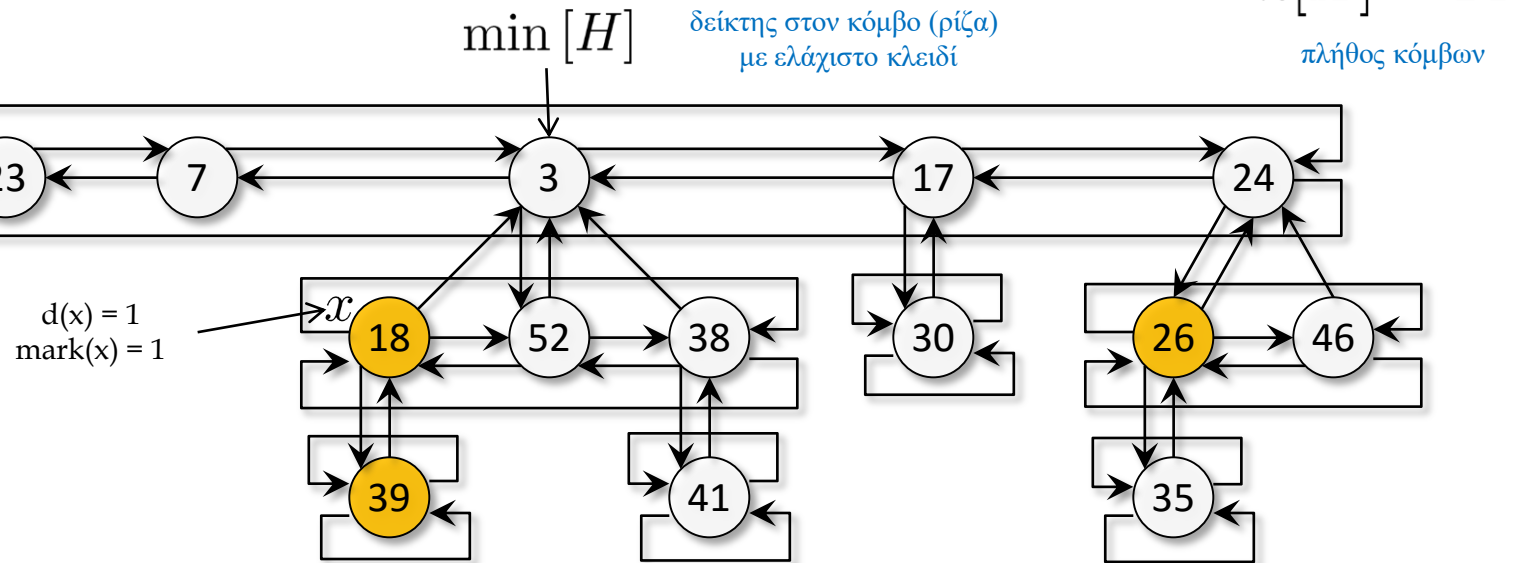
$$n[H] = 14$$



Η σωρός Fibonacci είναι πιο χαλαρή δομή από την διωνυμική σωρό

Αναπαράσταση Ουράς Fibonacci

$$n[H] = 14$$

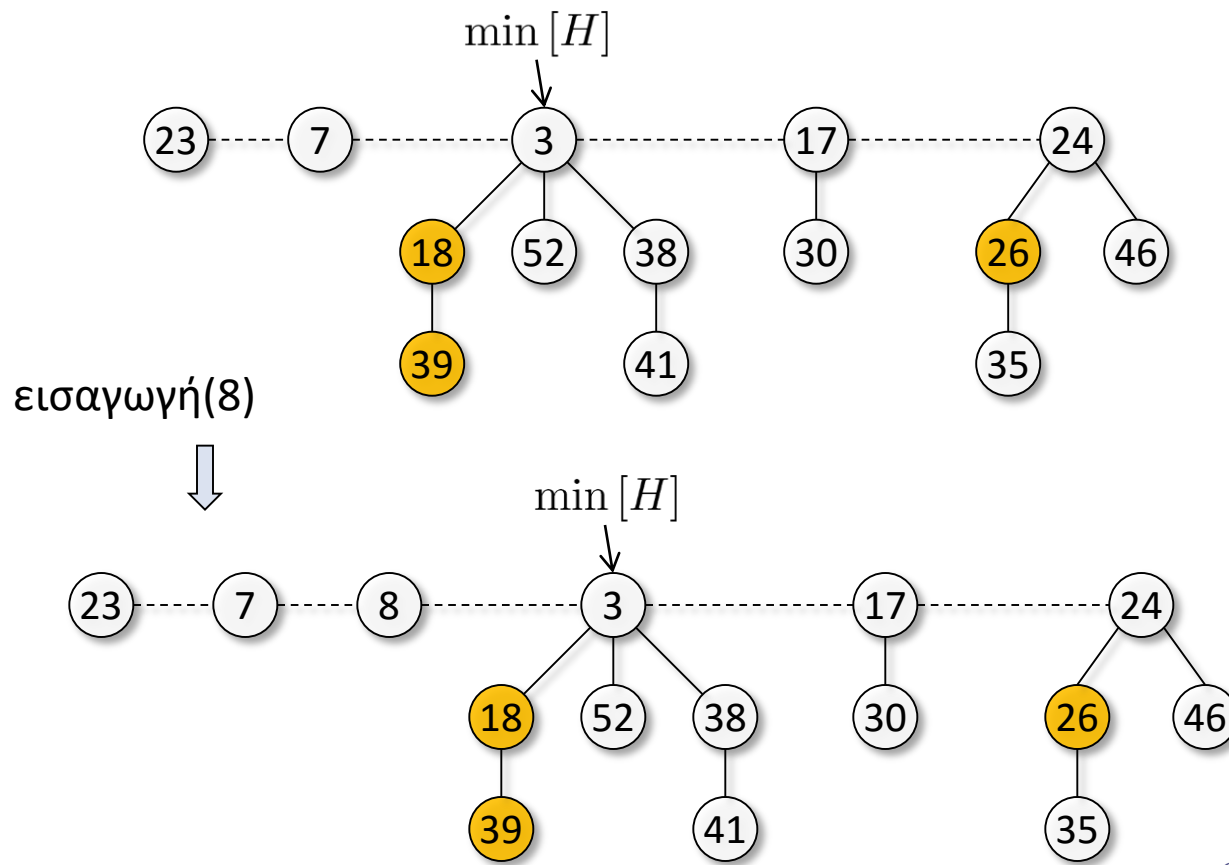


Η ουράς Fibonacci είναι πιο χαλαρή δομή από την διωνυμική ουρά

Πράξεις σε σωρούς Fibonacci

- ◉ Όταν θέλουμε να ενώσουμε 2 σωρούς Fibonacci (Meld) ενώνουμε τις λίστες ριζών και καθορίζουμε τον δείκτη ελαχίστου στη νέα δομή που δημιουργήθηκε.
- ◉ Όταν θέλουμε να εισάγουμε ένα στοιχείο σε ένα σωρό δημιουργούμε ένα διωνυμικό δέντρο για αυτό το στοιχείο μόνο και μετά το συνενώνουμε με το σωρό Fibonacci που έχουμε ήδη.
- ◉ **Επανορθωτικές πράξεις εκτελούμε όταν διαγράψουμε κάποιο στοιχείο από τον σωρό.**

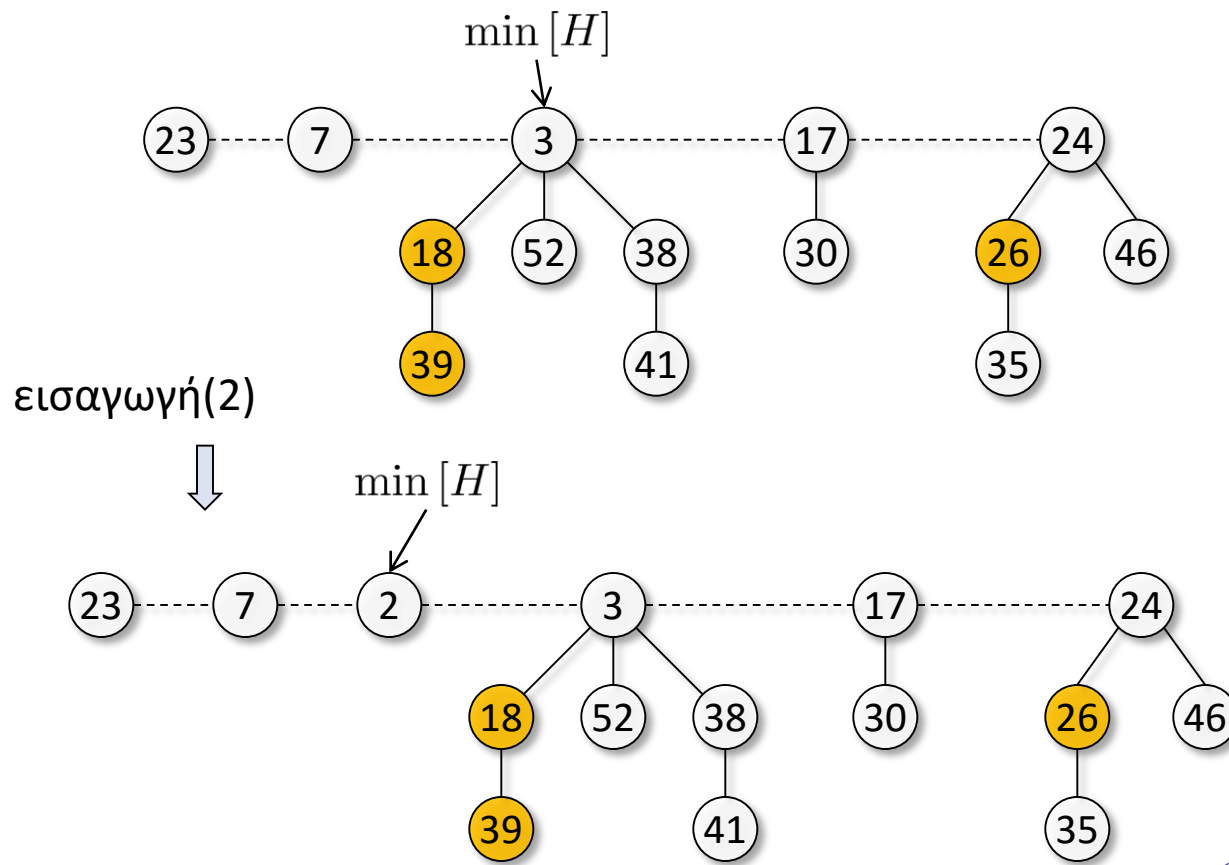
Εισαγωγή κόμβου σε σωρό Fibonacci



Δημιουργείται νέο δένδρο με μόνο ένα κόμβο και εισάγεται στη λίστα των ριζών δίπλα από το $\min[H]$

Αν το εισαγόμενο κλειδί είναι το ελάχιστο τότε ο δείκτης $\min[H]$ δείχνει στο νέο κόμβο

Εισαγωγή κόμβου σε σωρό Fibonacci

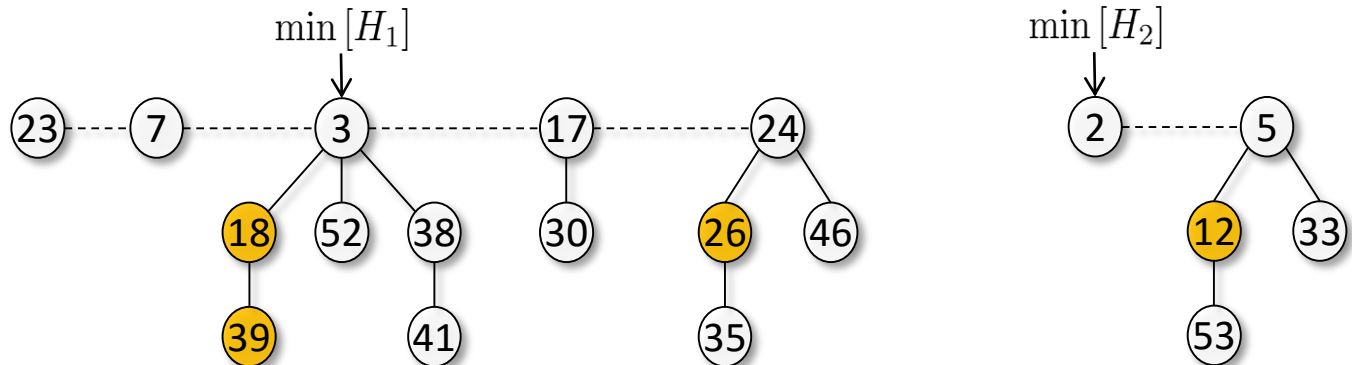


Δημιουργείται νέο δένδρο με μόνο ένα κόμβο και εισάγεται στη λίστα των ριζών δίπλα από το $\min[H]$

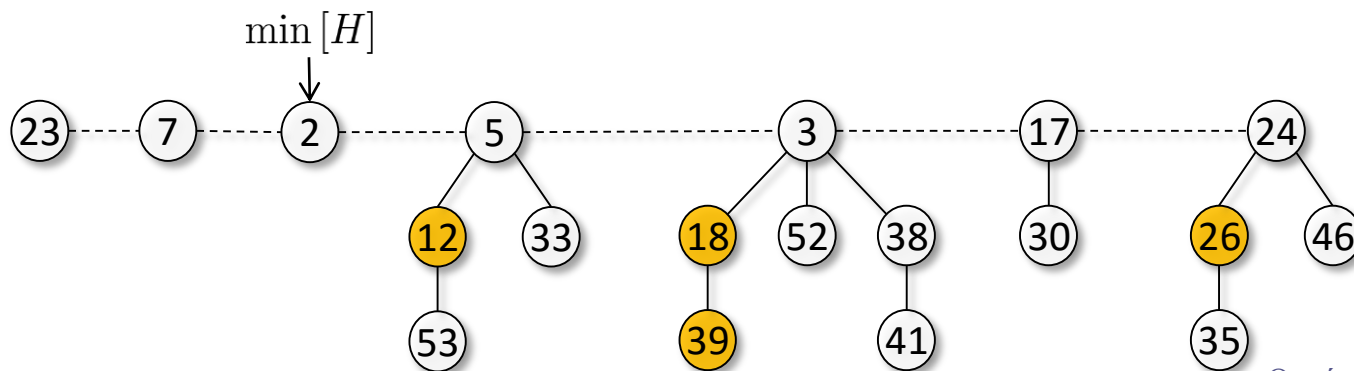
Αν το εισαγόμενο κλειδί είναι το ελάχιστο τότε ο δείκτης $\min[H]$ δείχνει στο νέο κόμβο

Ουρές Προτεραιότητας

Ένωση δύο σωρών Fibonacci



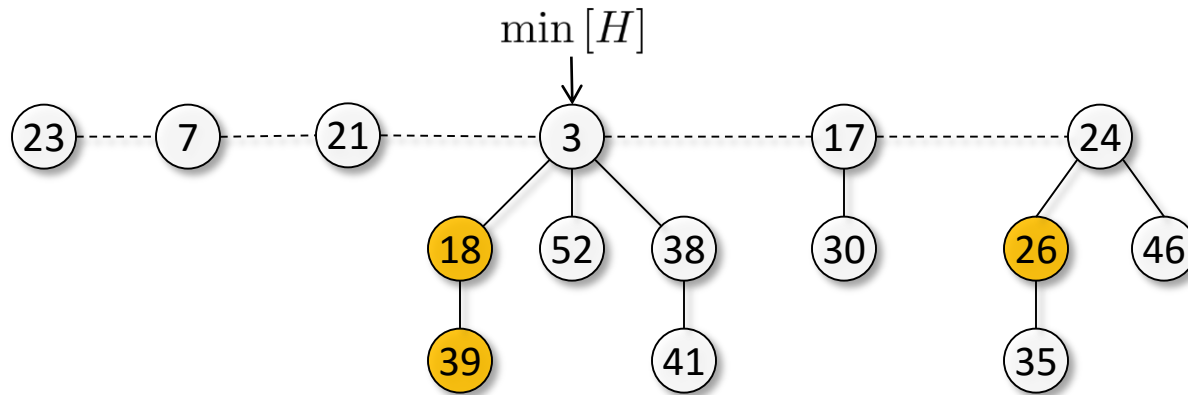
Ενώνει τις αντίστοιχες λίστες ριζικών κόμβων χρησιμοποιώντας τους δείκτες $\min[H_1]$ και $\min[H_2]$



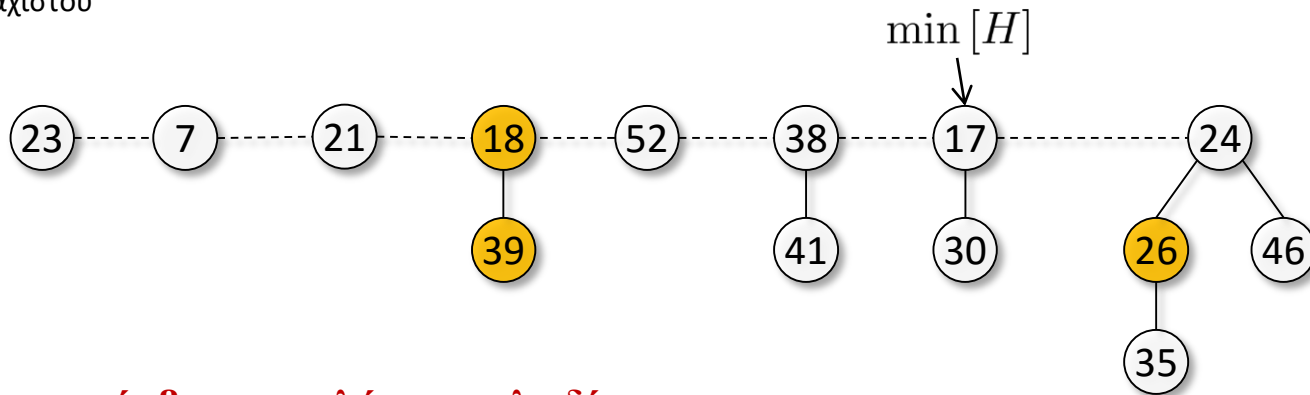
Ο δείκτης δείχνει στον κόμβο με το ελάχιστο κλειδί μεταξύ των $\min[H_1]$ και $\min[H_2]$

Ουρές Προτεραιότητας

Εξαγωγή ελάχιστου σωρού Fibonacci



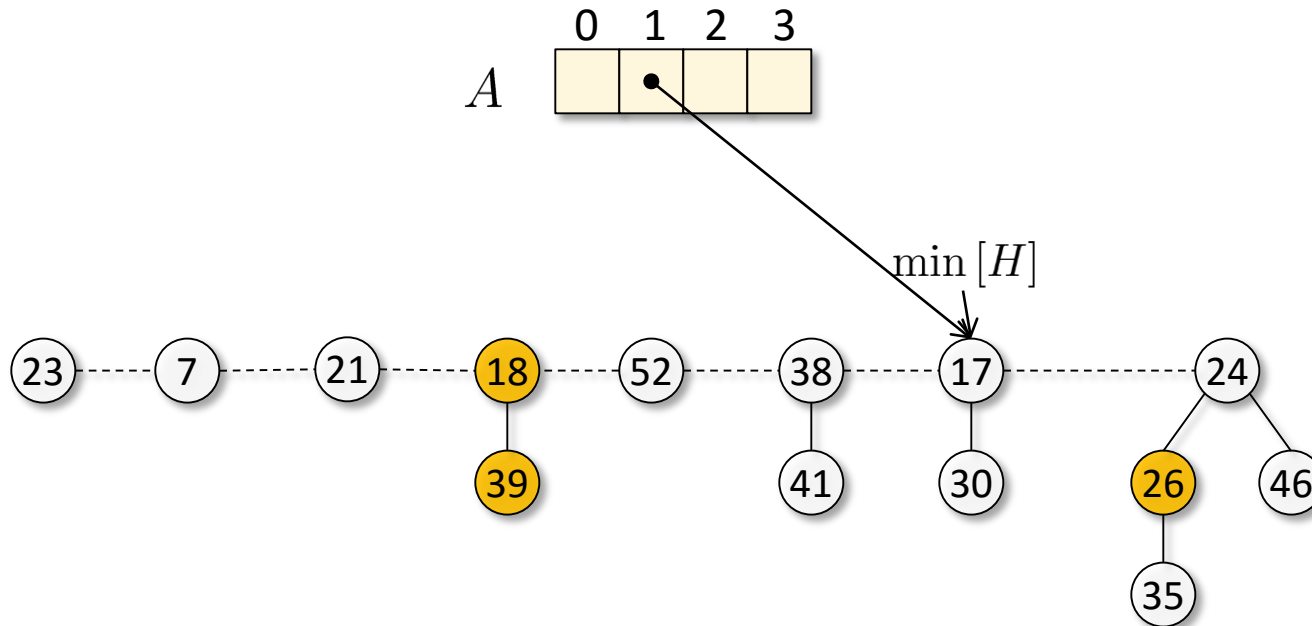
1^ο βήμα διαγραφής
ελάχιστου



**Διαγράφει τον κόμβο με το ελάχιστο κλειδί και
ενοποιεί δένδρα στο ριζικό επίπεδο**

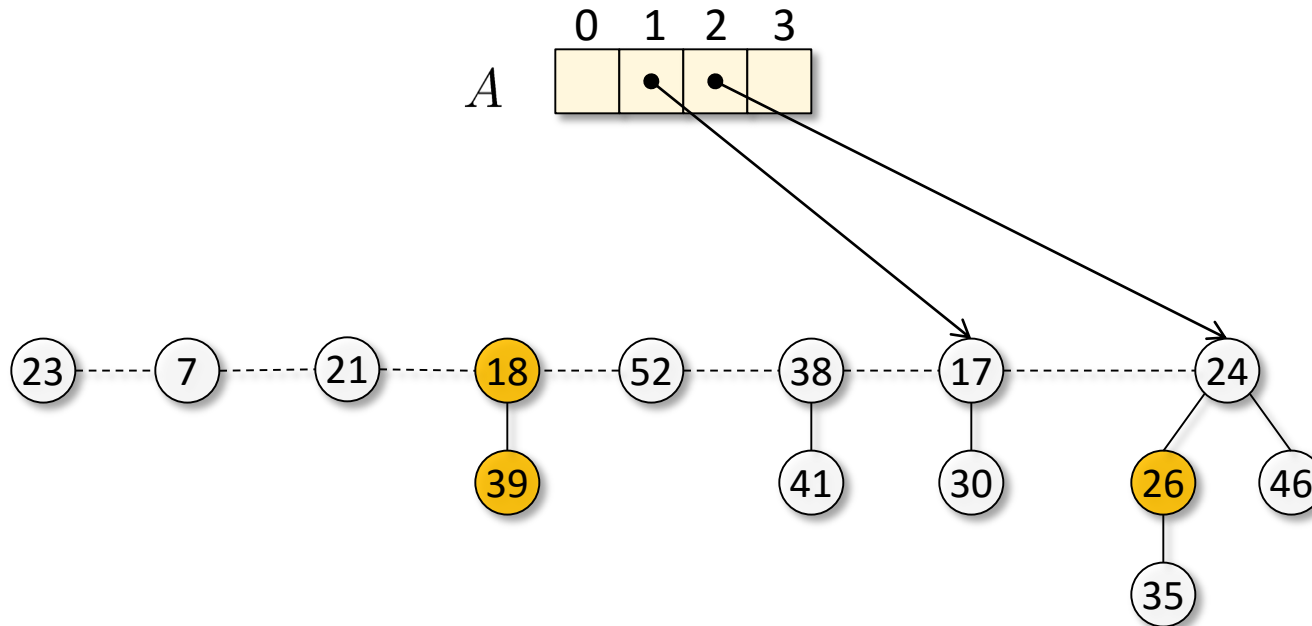
Εξαγωγή ελάχιστου σωρού Fibonacci

Η ρουτίνα ενοποίησης χρησιμοποιεί ένα βοηθητικό πίνακα δεικτών



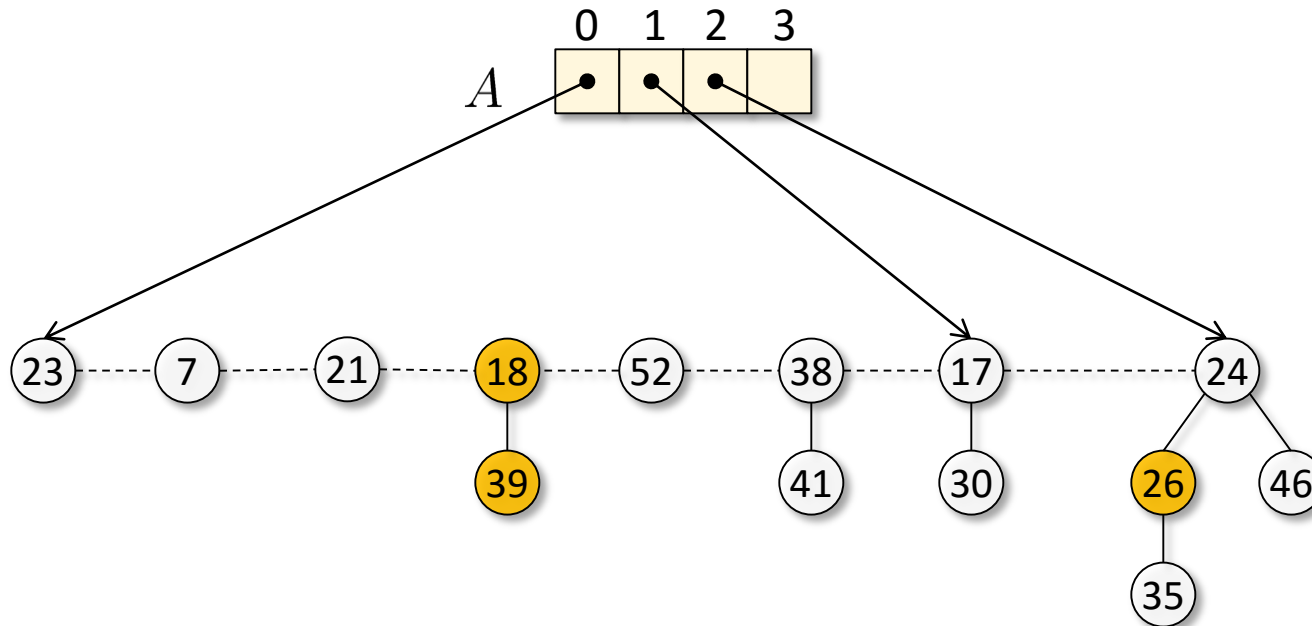
Εξαγωγή ελάχιστου σωρού Fibonacci

Η ρουτίνα ενοποίησης χρησιμοποιεί ένα βοηθητικό πίνακα δεικτών



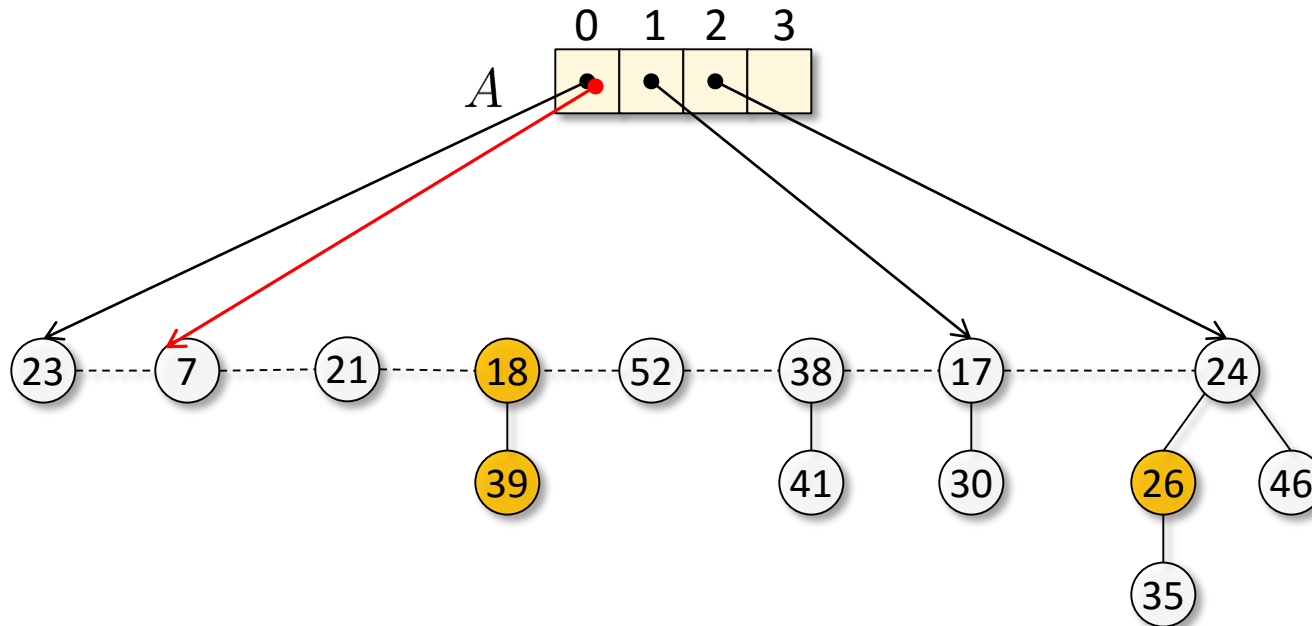
Εξαγωγή ελάχιστου σωρού Fibonacci

Η ρουτίνα ενοποίησης χρησιμοποιεί ένα βοηθητικό πίνακα δεικτών



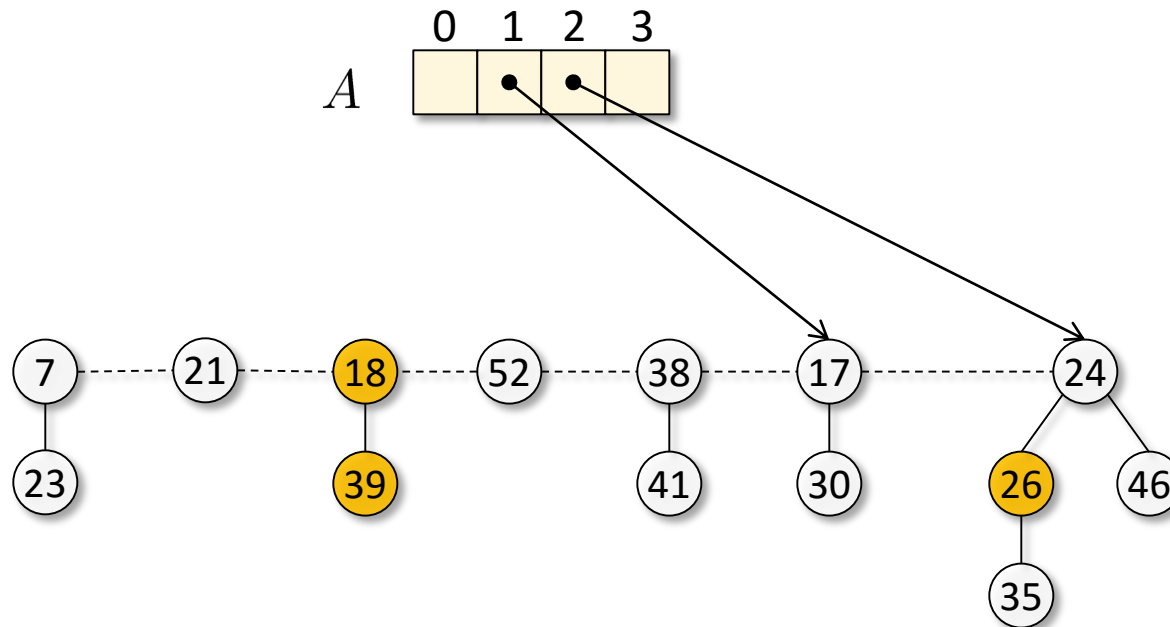
Εξαγωγή ελάχιστου σωρού Fibonacci

Η ρουτίνα ενοποίησης χρησιμοποιεί ένα βοηθητικό πίνακα δεικτών



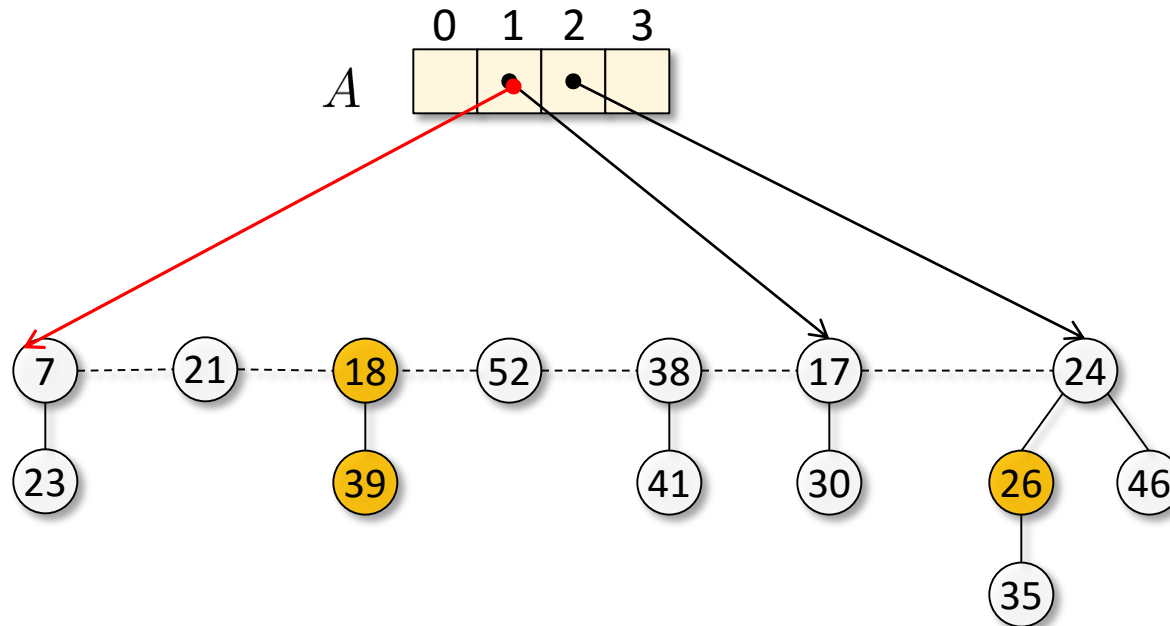
Εξαγωγή ελάχιστου σωρού Fibonacci

Η ρουτίνα ενοποίησης χρησιμοποιεί ένα βοηθητικό πίνακα δεικτών



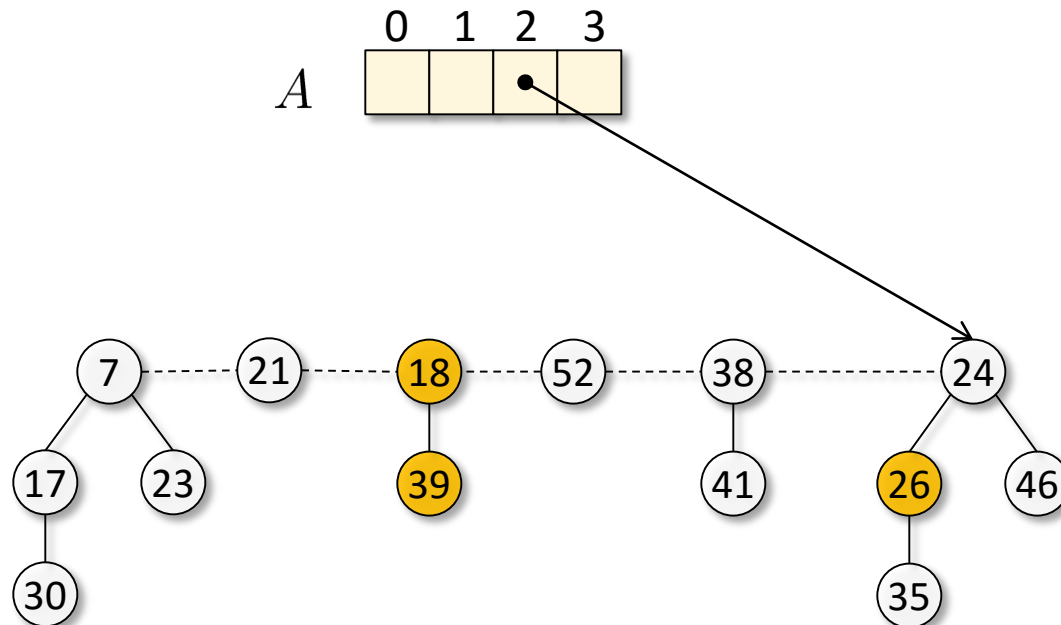
Εξαγωγή ελάχιστου σωρού Fibonacci

Η ρουτίνα ενοποίησης χρησιμοποιεί ένα βοηθητικό πίνακα δεικτών



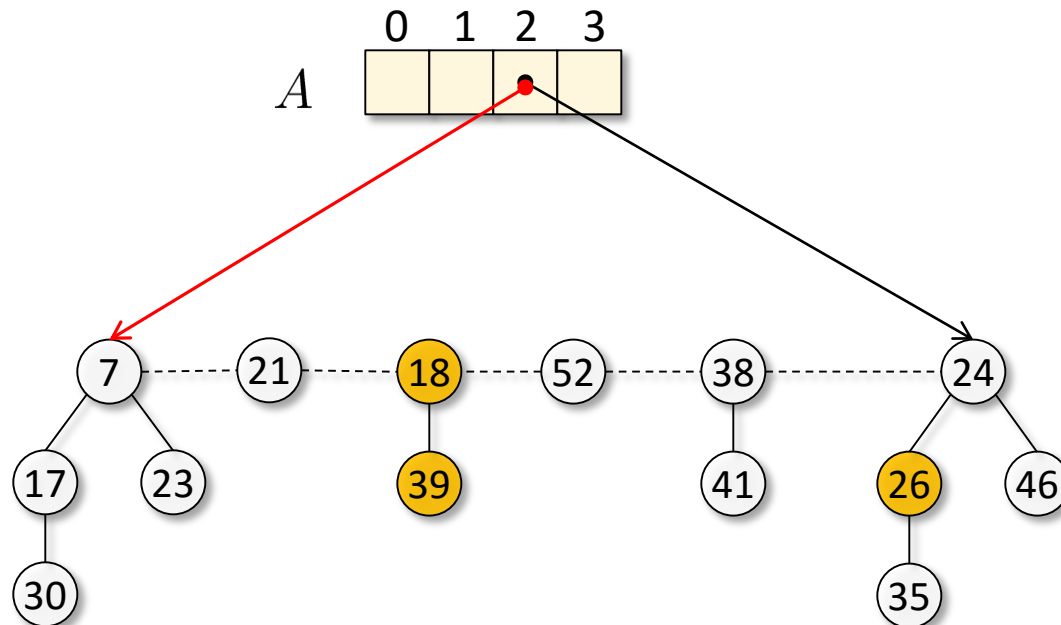
Εξαγωγή ελάχιστου σωρού Fibonacci

Η ρουτίνα ενοποίησης χρησιμοποιεί ένα βοηθητικό πίνακα δεικτών



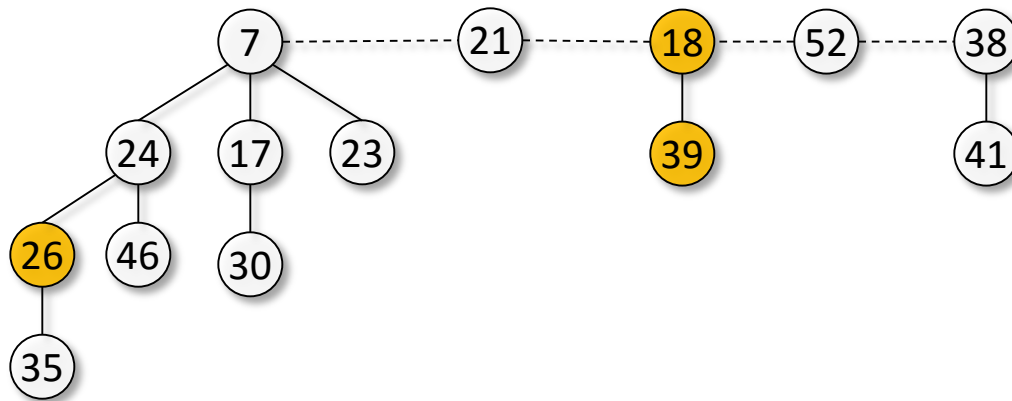
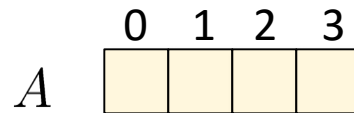
Εξαγωγή ελάχιστου σωρού Fibonacci

Η ρουτίνα ενοποίησης χρησιμοποιεί ένα βοηθητικό πίνακα δεικτών



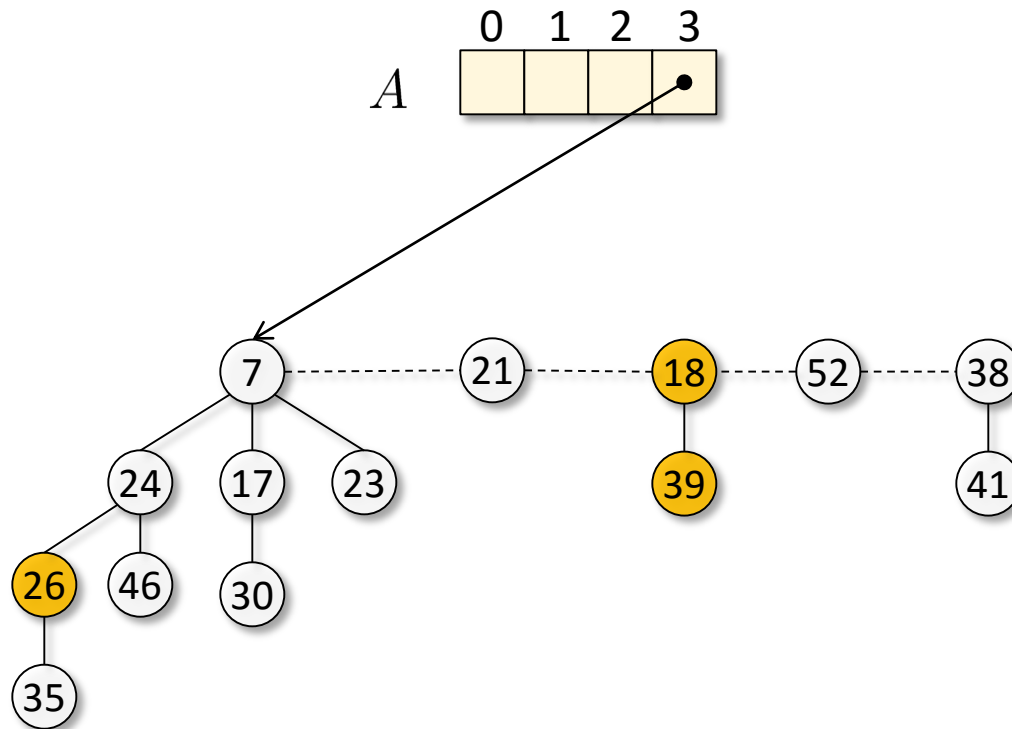
Εξαγωγή ελάχιστου σωρού Fibonacci

Η ρουτίνα ενοποίησης χρησιμοποιεί ένα βοηθητικό πίνακα δεικτών



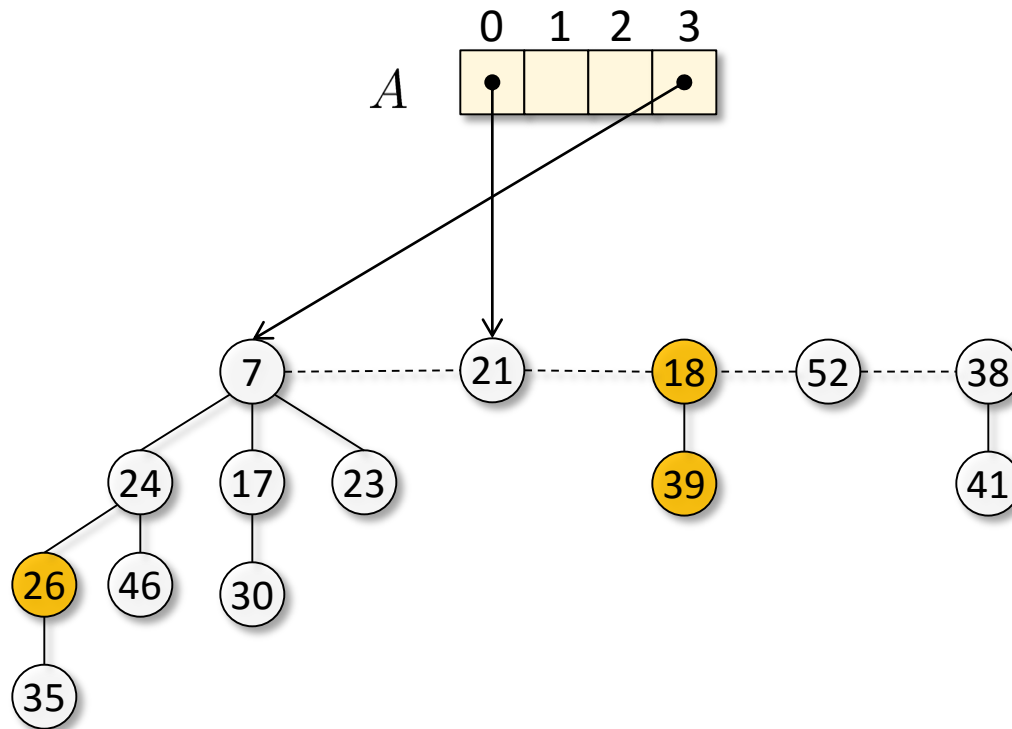
Εξαγωγή ελάχιστου σωρού Fibonacci

Η ρουτίνα ενοποίησης χρησιμοποιεί ένα βοηθητικό πίνακα δεικτών



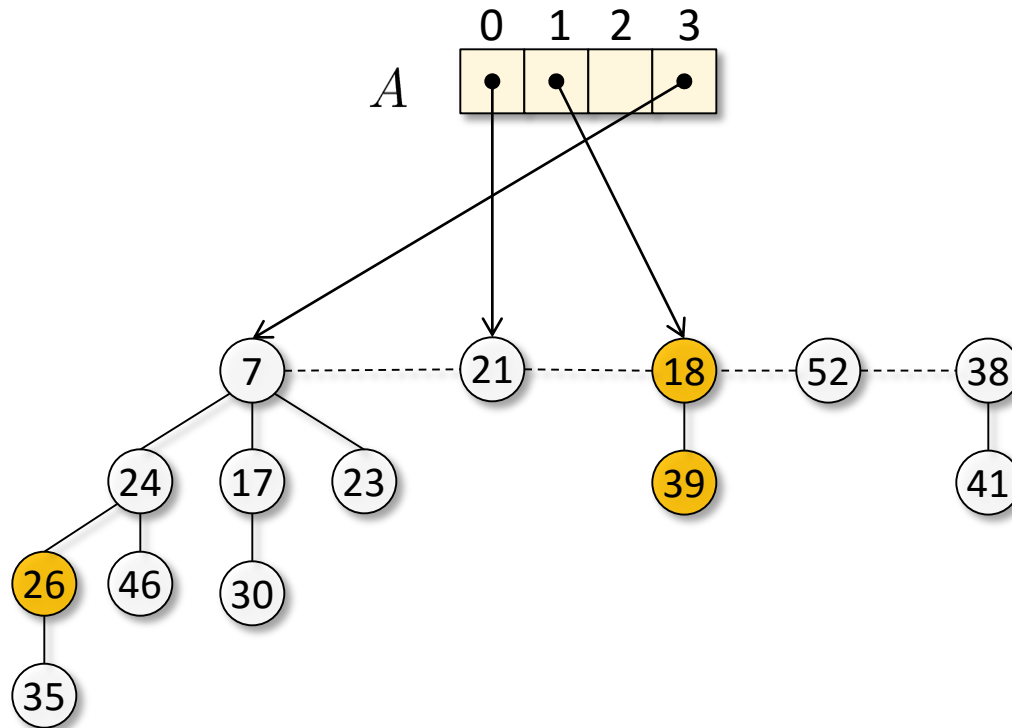
Εξαγωγή ελάχιστου σωρού Fibonacci

Η ρουτίνα ενοποίησης χρησιμοποιεί ένα βοηθητικό πίνακα δεικτών



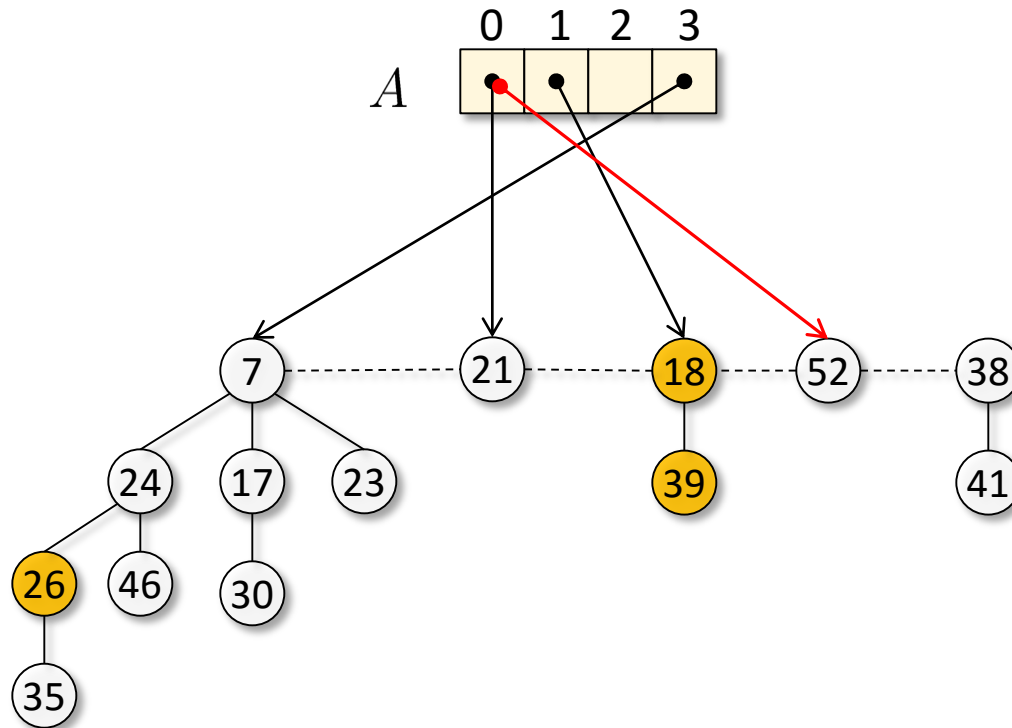
Εξαγωγή ελάχιστου σωρού Fibonacci

Η ρουτίνα ενοποίησης χρησιμοποιεί ένα βοηθητικό πίνακα δεικτών



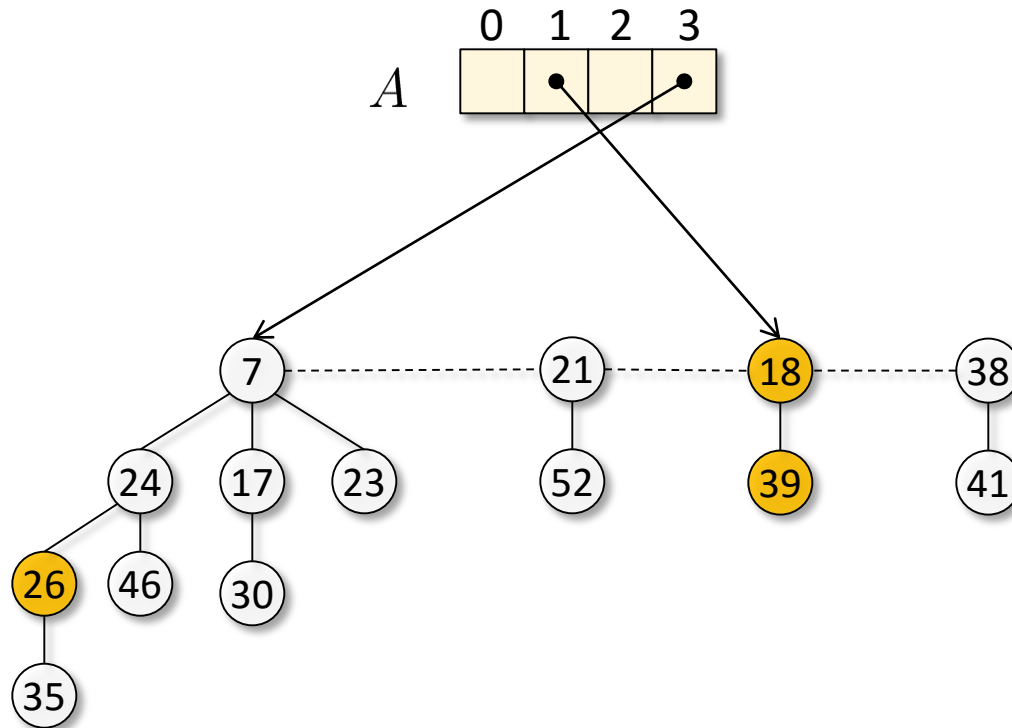
Εξαγωγή ελάχιστου σωρού Fibonacci

Η ρουτίνα ενοποίησης χρησιμοποιεί ένα βοηθητικό πίνακα δεικτών



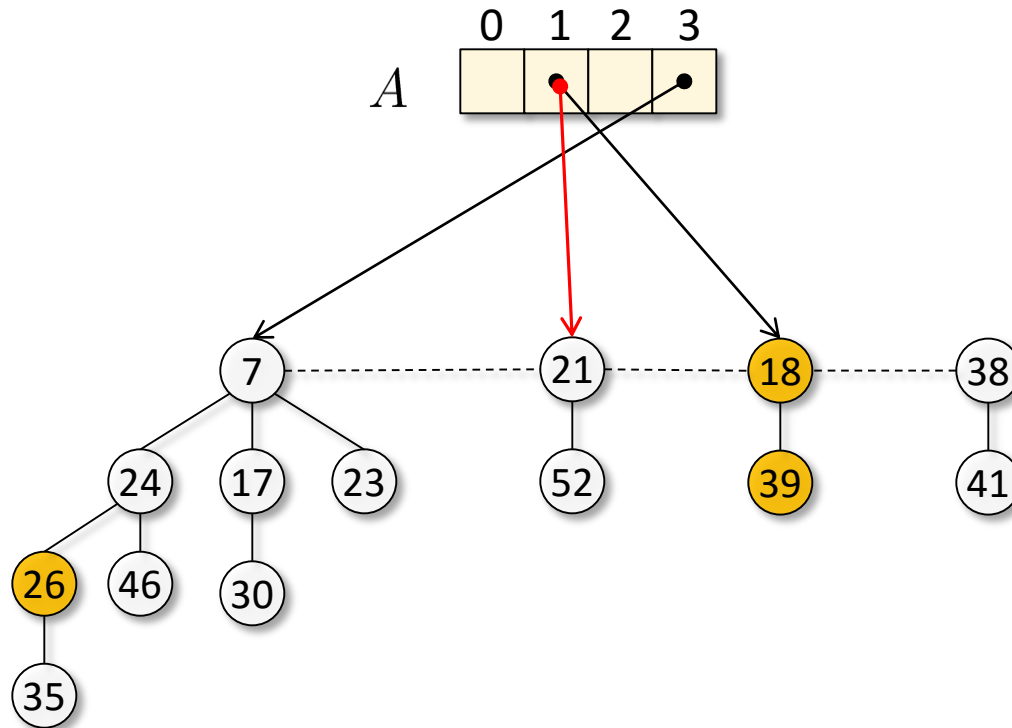
Εξαγωγή ελάχιστου σωρού Fibonacci

Η ρουτίνα ενοποίησης χρησιμοποιεί ένα βοηθητικό πίνακα δεικτών



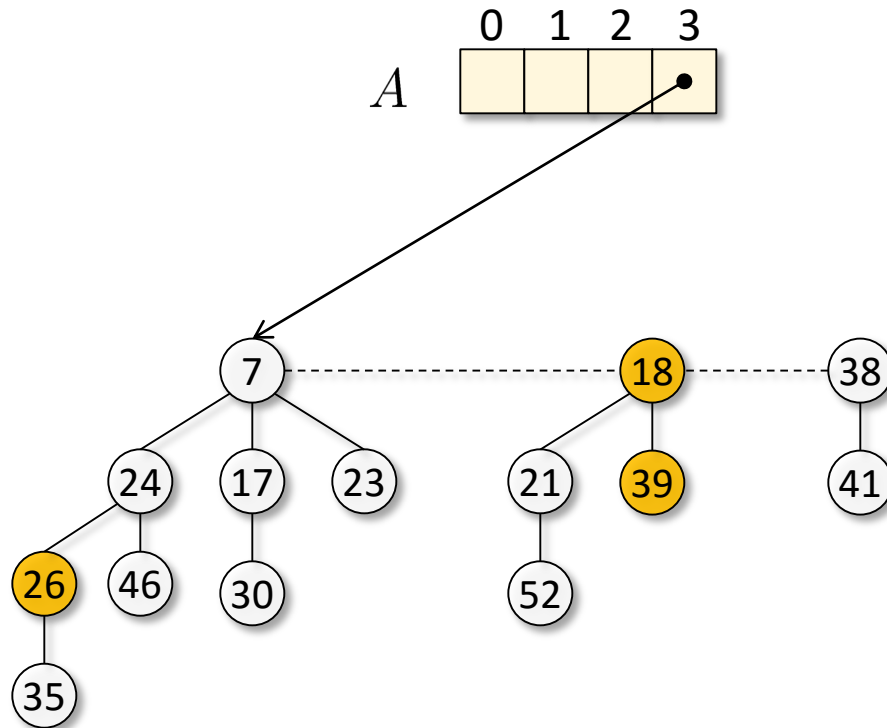
Εξαγωγή ελάχιστου σωρού Fibonacci

Η ρουτίνα ενοποίησης χρησιμοποιεί ένα βοηθητικό πίνακα δεικτών



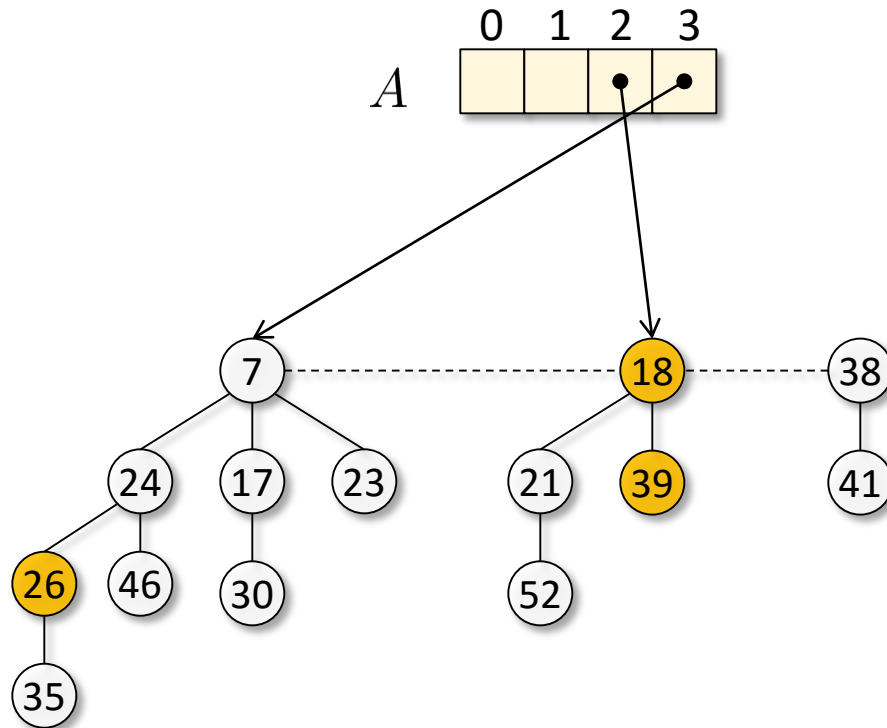
Εξαγωγή ελάχιστου σωρού Fibonacci

Η ρουτίνα ενοποίησης χρησιμοποιεί ένα βοηθητικό πίνακα δεικτών



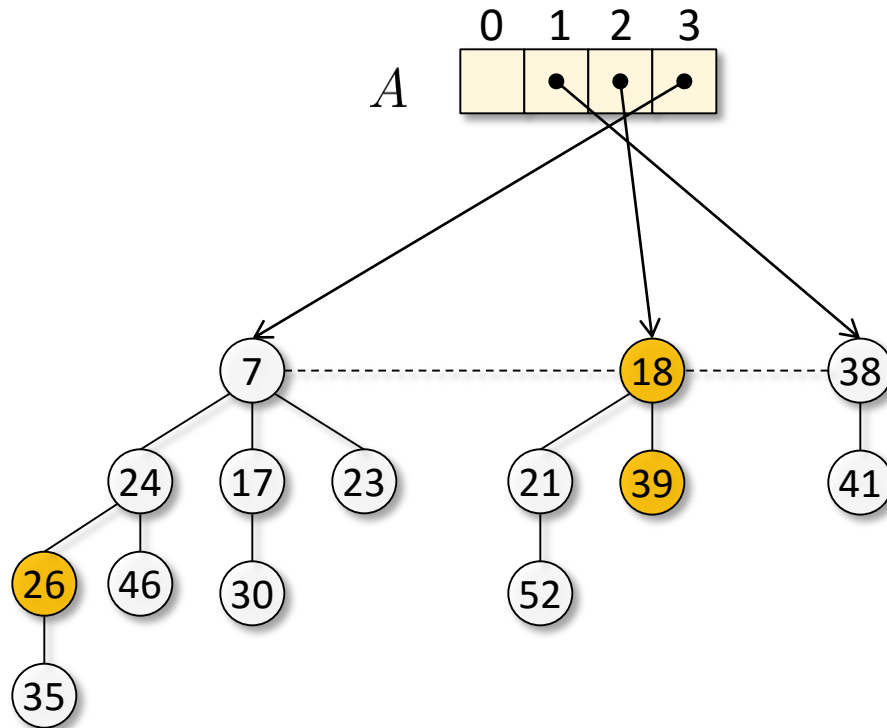
Εξαγωγή ελάχιστου σωρού Fibonacci

Η ρουτίνα ενοποίησης χρησιμοποιεί ένα βοηθητικό πίνακα δεικτών



Εξαγωγή ελάχιστου σωρού Fibonacci

Η ρουτίνα ενοποίησης χρησιμοποιεί ένα βοηθητικό πίνακα δεικτών



Εξαγωγή ελάχιστου σωρού Fibonacci

