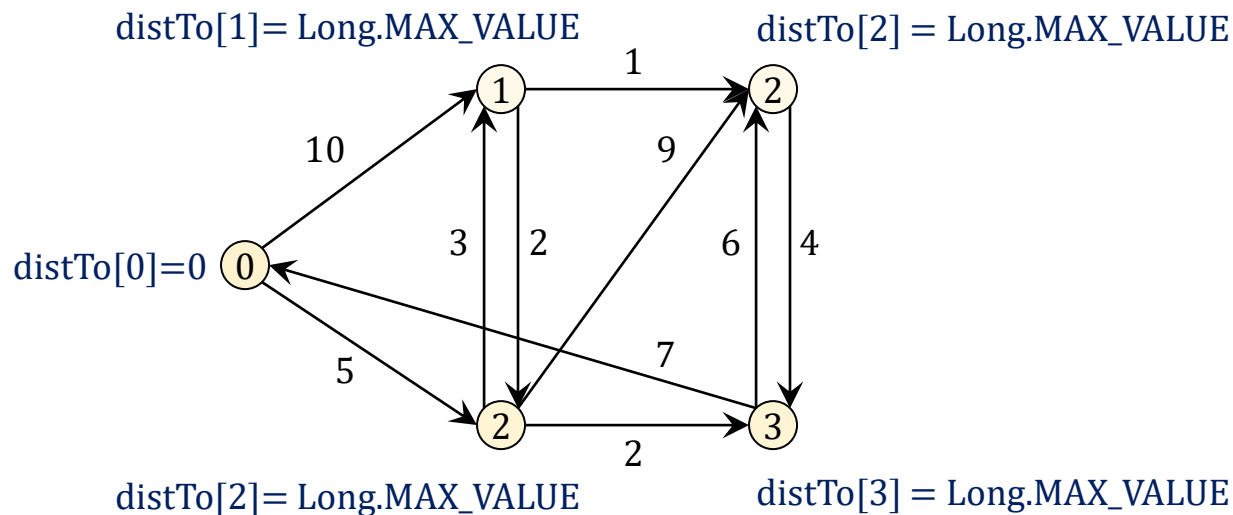


Δυναδικός Σωρός με Ευρετήριο

Ο αλγόριθμος του Dijkstra χρησιμοποιεί μια ουρά προτεραιότητας ελάχιστου όπου τα αντικείμενα είναι οι κόμβοι του γραφήματος και τα κλειδιά τους οι αποστάσεις από την αφετηρία

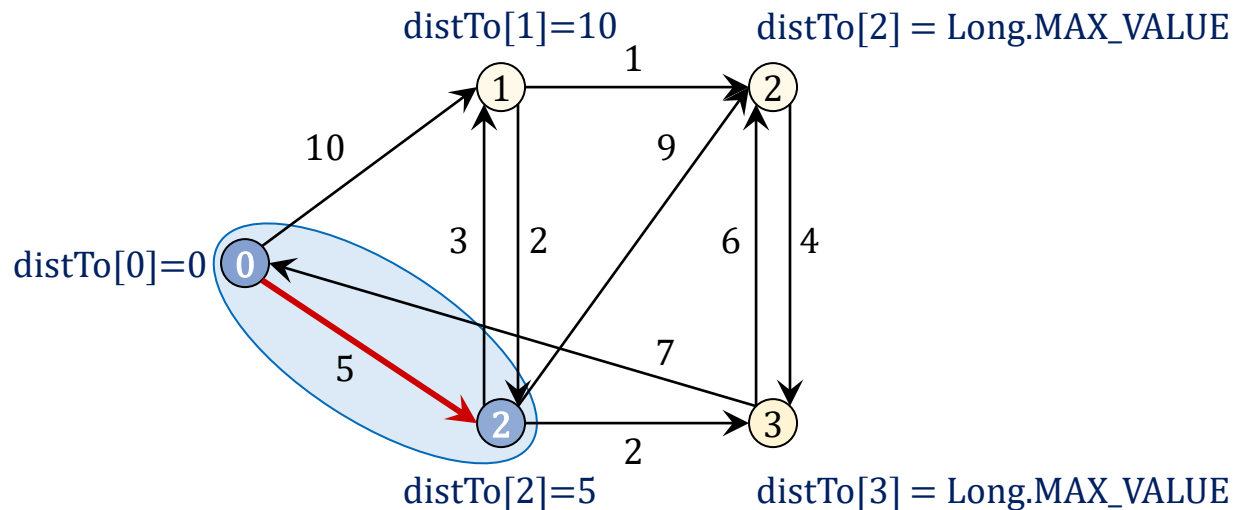
Αρχικοποίηση με αφετηρία τον κόμβο 0 ($\text{distTo}[v]$ =απόσταση του κόμβου v από την αφετηρία)



Δυναδικός Σωρός με Ευρετήριο

Ο αλγόριθμος του Dijkstra χρησιμοποιεί μια ουρά προτεραιότητας ελάχιστου όπου τα αντικείμενα είναι οι κόμβοι του γραφήματος και τα κλειδιά τους οι αποστάσεις από την αφετηρία

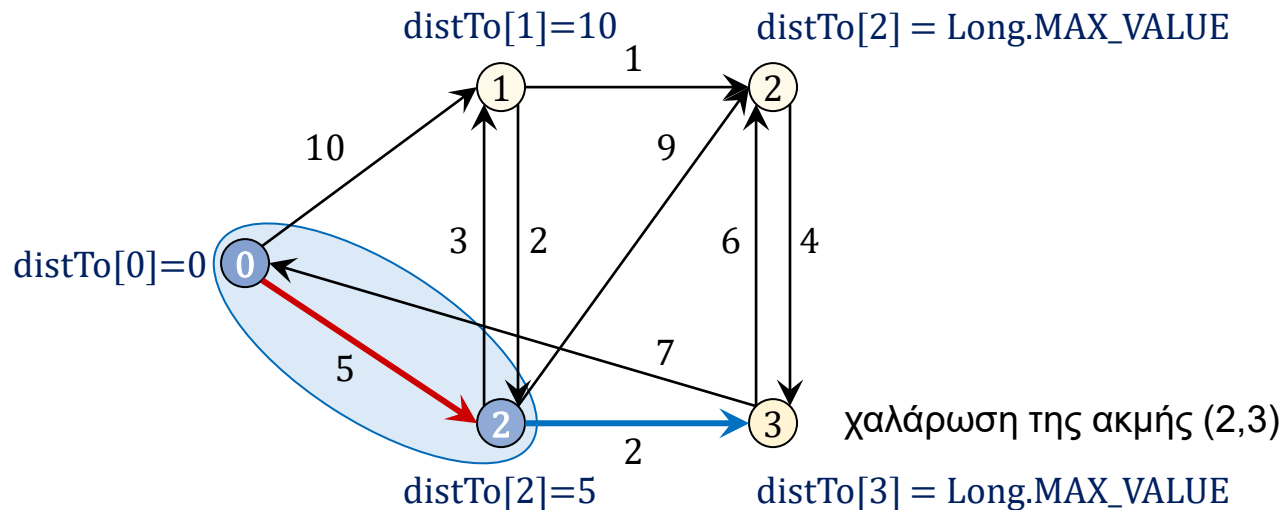
Μετά την εξαγωγή του κόμβου 2



Δυναδικός Σωρός με Ευρετήριο

Ο αλγόριθμος του Dijkstra χρησιμοποιεί μια ουρά προτεραιότητας ελάχιστου όπου τα αντικείμενα είναι οι κόμβοι του γραφήματος και τα κλειδιά τους οι αποστάσεις από την αφετηρία

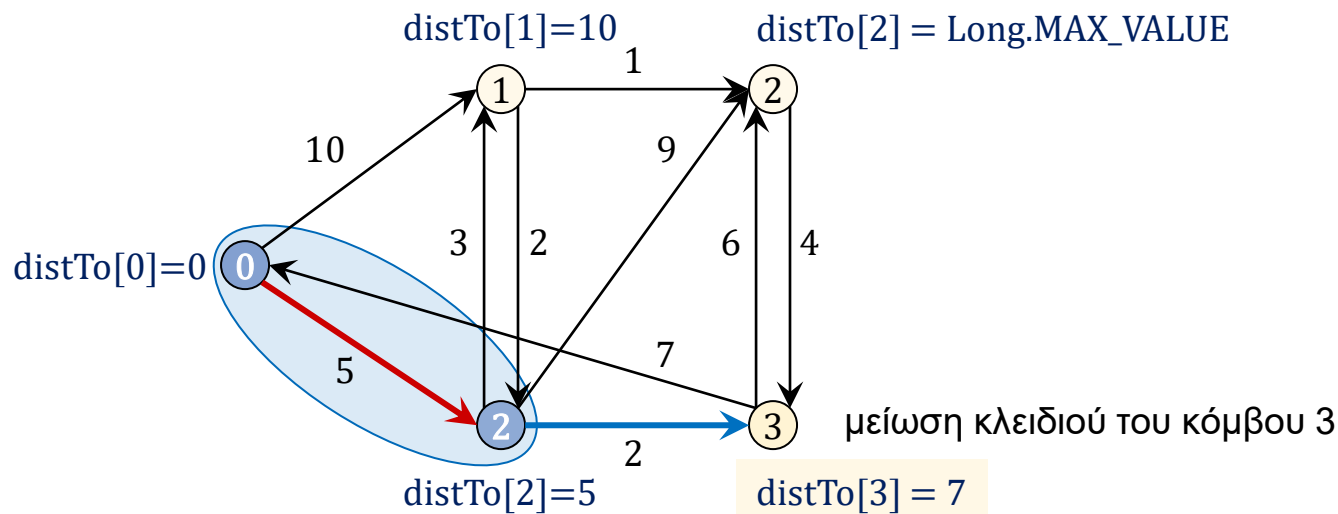
Μετά την εξαγωγή του κόμβου 2



Δυναδικός Σωρός με Ευρετήριο

Ο αλγόριθμος του Dijkstra χρησιμοποιεί μια ουρά προτεραιότητας ελάχιστου όπου τα αντικείμενα είναι οι κόμβοι του γραφήματος και τα κλειδιά τους οι αποστάσεις από την αφετηρία

Μετά την εξαγωγή του κόμβου 2



Δυναδικός Σωρός με Ευρετήριο

Για να υποστηρίξουμε τη λειτουργία **μείωσης κλειδιού** (ή πιο γενικά την **αλλαγή κλειδιού**) ενός αντικειμένου πρέπει να μπορούμε να βρούμε γρήγορα που βρίσκεται το αντικείμενο στο σωρό.

Για το σκοπό αυτό χρησιμοποιούμε ένα **δυναδικό σωρό με ευρετήριο**:

- Αποθηκεύουμε αντικείμενα τα οποία έχουν ένα κλειδί (γενικού τύπου Key) και μια ακέραια ταυτότητα από 0 έως $N-1$.
- Χρησιμοποιούμε τρεις πίνακες: `keys[]`, `pq[]` και `index[]`:

`keys[j]` = κλειδί του αντικειμένου (με ταυτότητα) `j`

`pq[i]` = αντικείμενο στη θέση `i` του σωρού

`index[j]` = θέση του αντικειμένου (με ταυτότητα) `j` στο σωρό

Ισχύει `pq[index[j]] == index[pq[j]] == j`

Δυναδικός Σωρός με Ευρετήριο

Π.χ., μετά την εισαγωγή των αντικειμένων με ταυτότητες και αντίστοιχα κλειδιά

(0, 60), (1, 48), (2, 29), (3, 47), (4, 15),

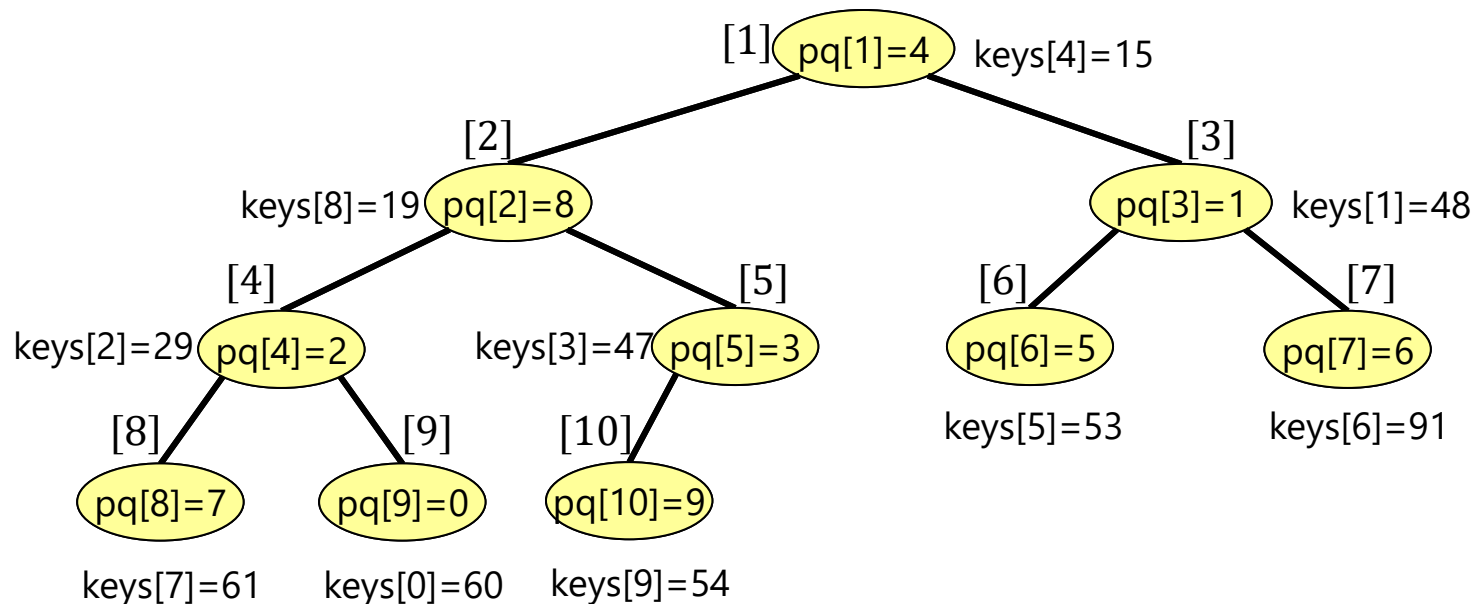
(5, 53), (6, 91), (7, 61), (8, 19), (9, 54)

θα έχουμε

keys[0:9] = [60, 48, 29, 47, 15, 53, 91, 61, 19, 54]

pq[1:10] = [4, 8, 1, 2, 3, 5, 6, 7, 0, 9]

index[0:9] = [9, 3, 4, 5, 1, 6, 7, 8, 2, 10]



Δυναδικός Σωρός με Ευρετήριο

Π.χ., μετά την εισαγωγή των αντικειμένων με ταυτότητες και αντίστοιχα κλειδιά

(0, 60), (1, 48), (2, 29), (3, 47), (4, 15),

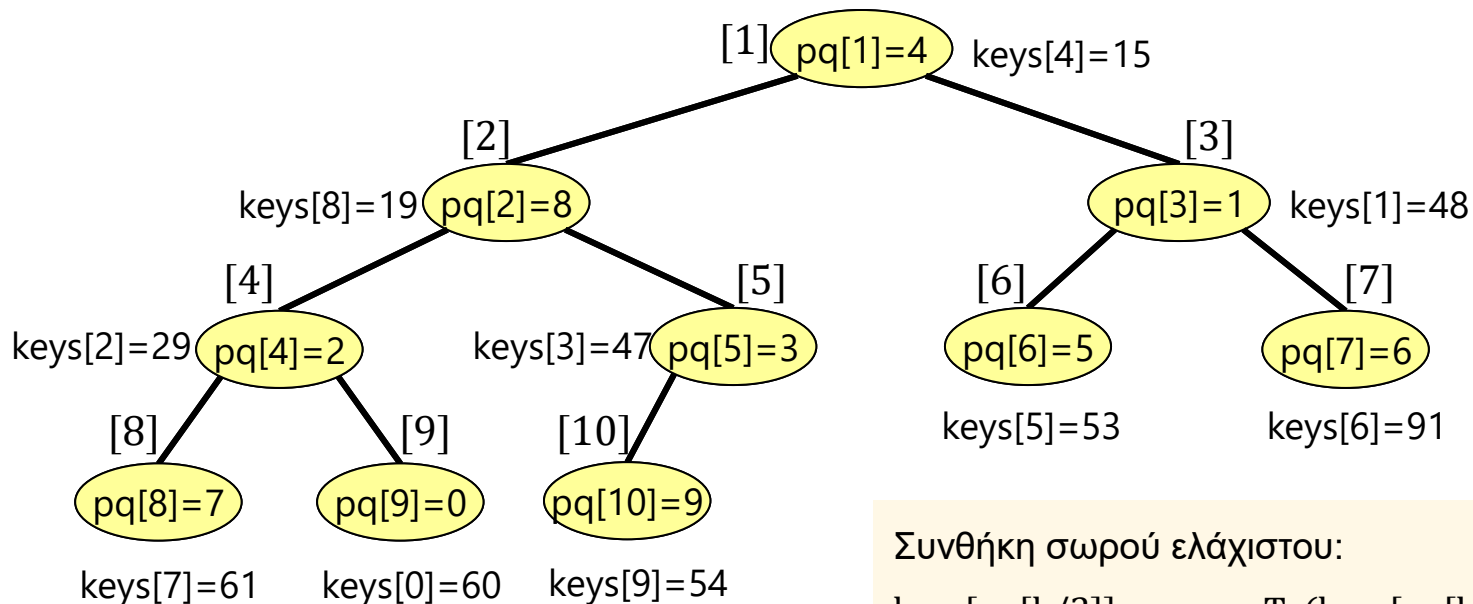
(5, 53), (6, 91), (7, 61), (8, 19), (9, 54)

θα έχουμε

keys[0:9] = [60, 48, 29, 47, 15, 53, 91, 61, 19, 54]

pq[1:10] = [4, 8, 1, 2, 3, 5, 6, 7, 0, 9]

index[0:9] = [9, 3, 4, 5, 1, 6, 7, 8, 2, 10]



Συνθήκη σωρού ελάχιστου:

$\text{keys}[\text{pq}[\text{k}/2]].\text{compareTo}(\text{keys}[\text{pq}[\text{k}]]) < 0$

Δυαδικός Σωρός με Ευρετήριο

Υλοποίηση **αλλαγής κλειδιού** του αντικειμένου (με ταυτότητα) **j**:

- Αποθηκεύουμε τη νέα τιμή **v** του κλειδιού του **j** στον πίνακα **keys**, θέτοντας **keys[j]=v**
- Βρίσκουμε τη θέση του αντικειμένου **j** στο δυαδικό σωρό: **k=index[j]**
- Αν το κλειδί του **j** έχει μειωθεί τότε εκτελούμε **fixUp(k)**
- Διαφορετικά, αν το κλειδί του **j** έχει αυξηθεί, τότε εκτελούμε **fixDown(k)**
- Όταν γίνεται αντιμετάθεση δύο αντικειμένων στον πίνακα **pq[]**, τότε θα πρέπει να ενημερώνουμε τις νέες θέσεις τους στον πίνακα **index[]**

Δυναδικός Σωρός με Ευρετήριο

Υλοποίηση **μείωσης κλειδιού** του αντικειμένου (με ταυτότητα) **j**:

```
// change the key of item j
public void change(int j, Key key) {
    Key oldKey = keys[j];
    keys[j] = key;
    int k = index[j];
    if ( key.compareTo(oldKey) < 0 )
        fixUp(k);
    else
        fixDown(k);
}
```

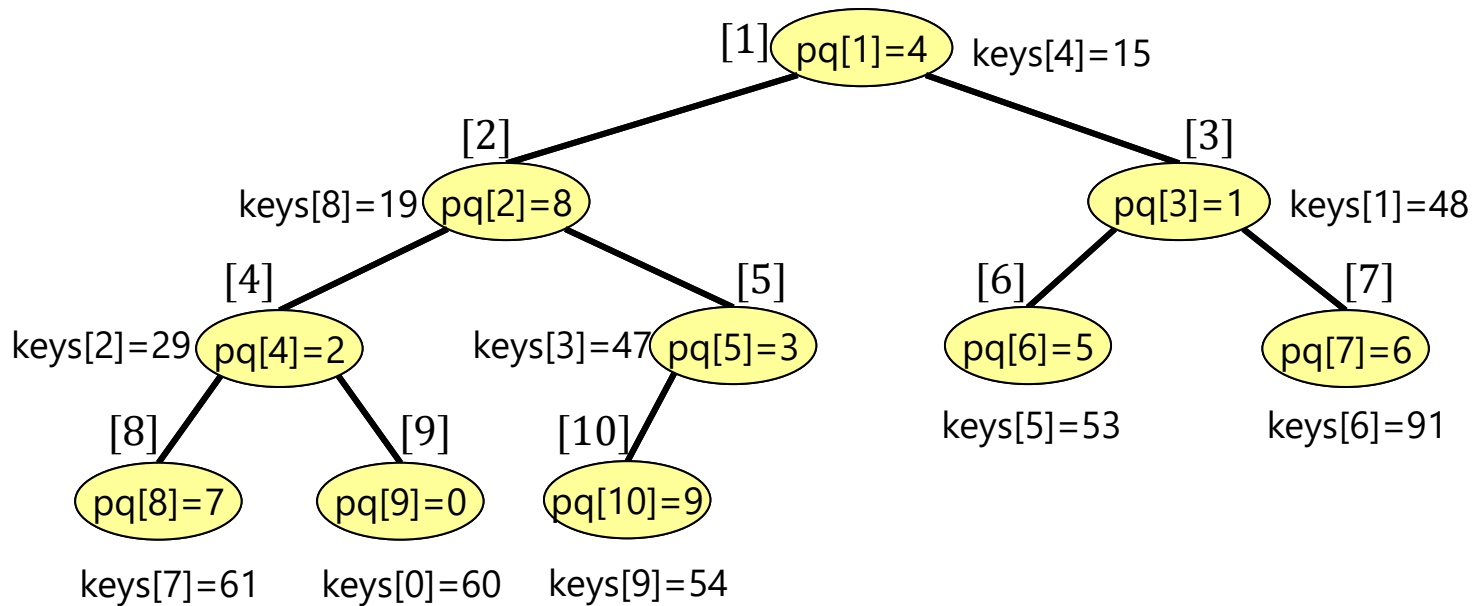
Δυαδικός Σωρός με Ευρετήριο

Παράδειγμα μείωσης κλειδιού

keys[0:9] = [60, 48, 29, 47, 15, 53, 91, 61, 19, 54]

pq[1:10] = [4, 8, 1, 2, 3, 5, 6, 7, 0, 9]

index[0:9] = [9, 3, 4, 5, 1, 6, 7, 8, 2, 10]



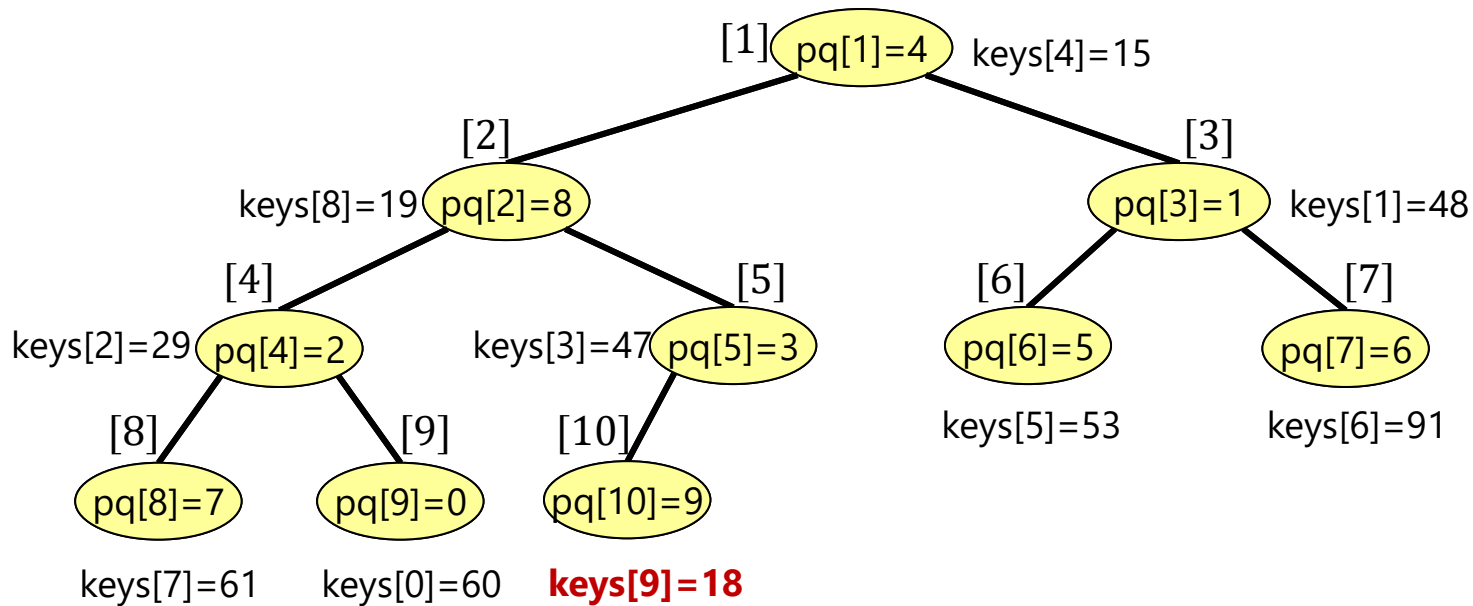
Δυαδικός Σωρός με Ευρετήριο

Παράδειγμα μείωσης κλειδιού

keys[0:9] = [60, 48, 29, 47, 15, 53, 91, 61, 19, **18**]

pq[1:10] = [4, 8, 1, 2, 3, 5, 6, 7, 0, **9**]

index[0:9] = [9, 3, 4, 5, 1, 6, 7, 8, 2, 10]



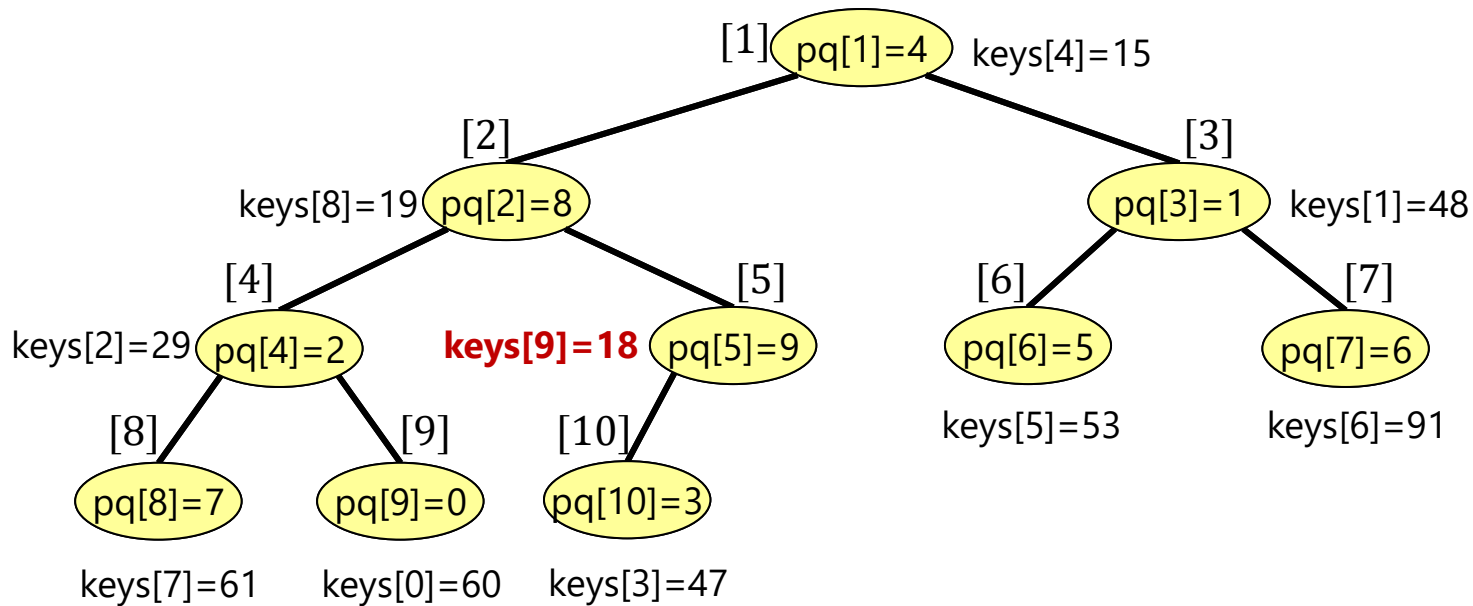
Δυαδικός Σωρός με Ευρετήριο

Παράδειγμα μείωσης κλειδιού

keys[0:9] = [60, 48, 29, 47, 15, 53, 91, 61, 19, **18**]

pq[1:10] = [4, 8, 1, 2, **9**, 5, 6, 7, 0, **3**]

index[0:9] = [9, 3, 4, **10**, 1, 6, 7, 8, 2, **5**]



Δυαδικός Σωρός με Ευρετήριο

Παράδειγμα μείωσης κλειδιού

keys[0:9] = [60, 48, 29, 47, 15, 53, 91, 61, 19, **18**]

pq[1:10] = [4, **9**, 1, 2, **8**, 5, 6, 7, 0, 3]

index[0:9] = [9, 3, 4, 10, 1, 6, 7, 8, **5**, **2**]

