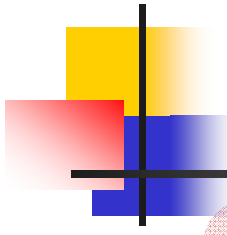
Quick Sort



- ***θ*** το πρώτο στοιχείο (οδηγός) του πίνακα στην οριστική του θέση
- **Διαίρεση** του πίνακα σε δύο ζώνες

  αρχή του πίνακα: στοιχεία ≤ ***θ***

  τέλος του πίνακα: στοιχεία >***θ***

- **Αναδρομική κλήση** του αλγόριθμου σε κάθε μία από τις ζώνες όσο δεν είναι ελαττωμένες σε ένα στοιχείο

Παράδειγμα - Οριστική θέση οδηγού

a: $\underline{\mathbf{6}}_1$ 9_2 2_3 7_4 4_5 5_6 8_7

a: x x x **6** x x x

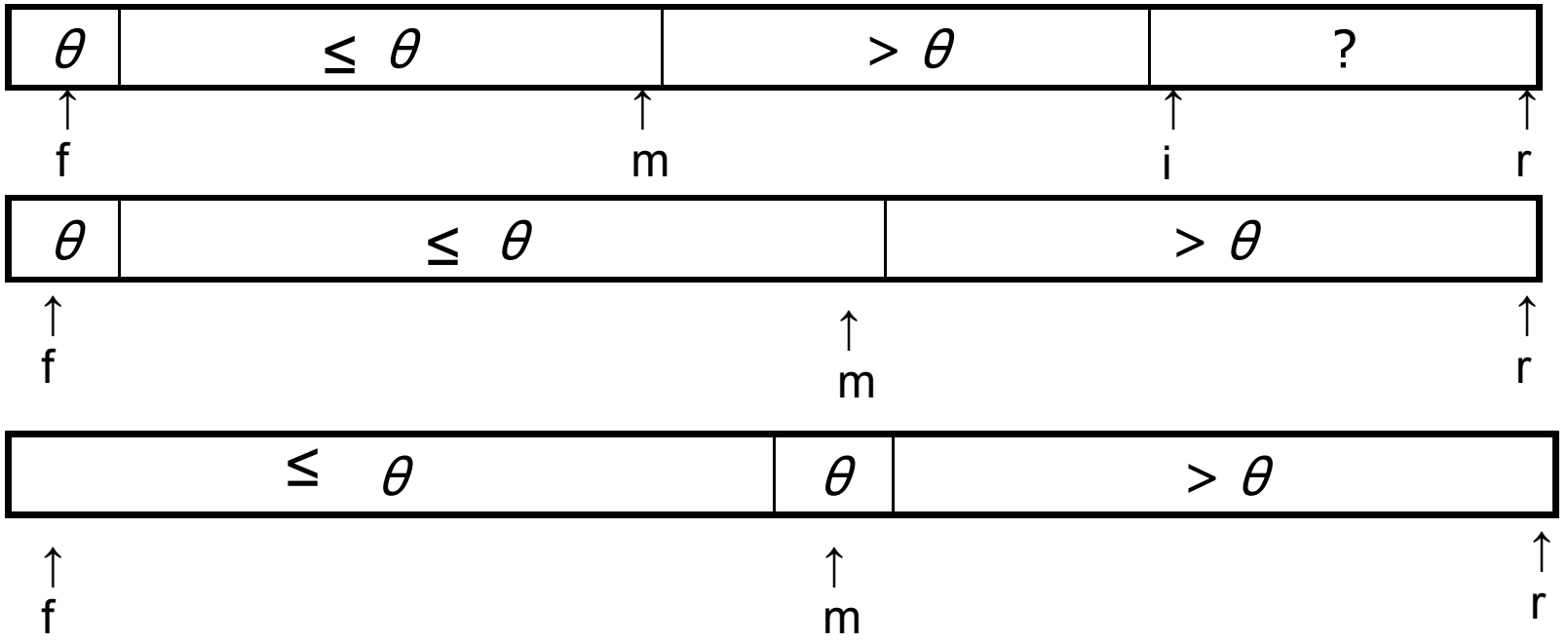
a: $\underline{2}_3$ 7_4 4_5 5_6

a: $\underline{2}_3$ x x x

"Εύρεση της τελικής θέσης του οδηγού στην αρχή του αλγόριθμου"

Quicksort: Οριστική θέση οδηγού θ

- Έστω f (αριστερός δείκτης) και r (δεξιός δείκτης)
- Έστω δύο δείκτες i και m τέτοιοι ώστε για κάθε στιγμή έχουμε: $\mathbf{a_j} \leq \boldsymbol{\theta}$, f < j ≤ m και $\mathbf{a_j} > \boldsymbol{\theta}$, m <j < i



Παράδειγμα - Οριστική θέση οδηγού

a: **6**$_1$ 9$_2$ 2$_3$ 7$_4$ 4$_5$ 5$_6$ 8$_7$

↓

↓

a: x x x **6** x x x

Quicksort: Διαμέριση

a: $\mathbf{6}_1$ $\underline{9}_2$ 2_3 7_4 4_5 5_6 8_7 (1)

a: $\mathbf{6}_1$ 9_2 $\underline{2}_3$ 7_4 4_5 5_6 8_7 (2)

a: $\mathbf{6}_1$ $\mathbf{2}_2$ 9_3 $\underline{7}_4$ 4_5 5_6 8_7 (3)

a: $\mathbf{6}_1$ $\mathbf{2}_2$ 9_3 7_4 $\underline{4}_5$ 5_6 8_7 (4)

a: $\mathbf{6}_1$ $\mathbf{2}_2$ $\mathbf{4}_3$ 7_4 9_5 $\underline{5}_6$ 8_7 (5)

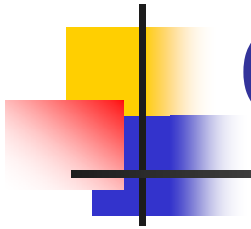
a: $\mathbf{6}_1$ $\mathbf{2}_2$ $\mathbf{4}_3$ $\mathbf{5}_4$ 9_5 7_6 $\underline{8}_7$ (6)

a: $\mathbf{6}_1$ $\mathbf{2}_2$ $\mathbf{4}_3$ $\mathbf{5}_4$ 9_5 7_6 8_7

↓

a: 5 2 4 **6** 9 7 8

Quicksort:Διαμέριση



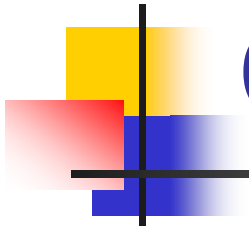
a: **5** 2 4 (1)

a: **5** **2** 4 (2)

a: **5** 2 **4**

↓

a: 4 2 **5**



Quicksort:Διαμέριση

a: **4** 2

a: **4** 2

↓

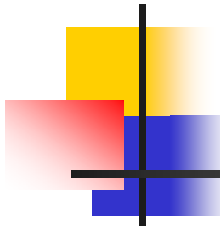
a: 2 **4**

Αλγόριθμος Partition

```
Partition (A, f, r)

θ = A[f], m = f
for i = f+1 to r do
      if A[i] ≤ θ
        m = m+1
        swap (A[i], A[m])
      end if
end for
swap (A[f], A[m])
return m /* οριστική θέση οδηγού θ */
```

****



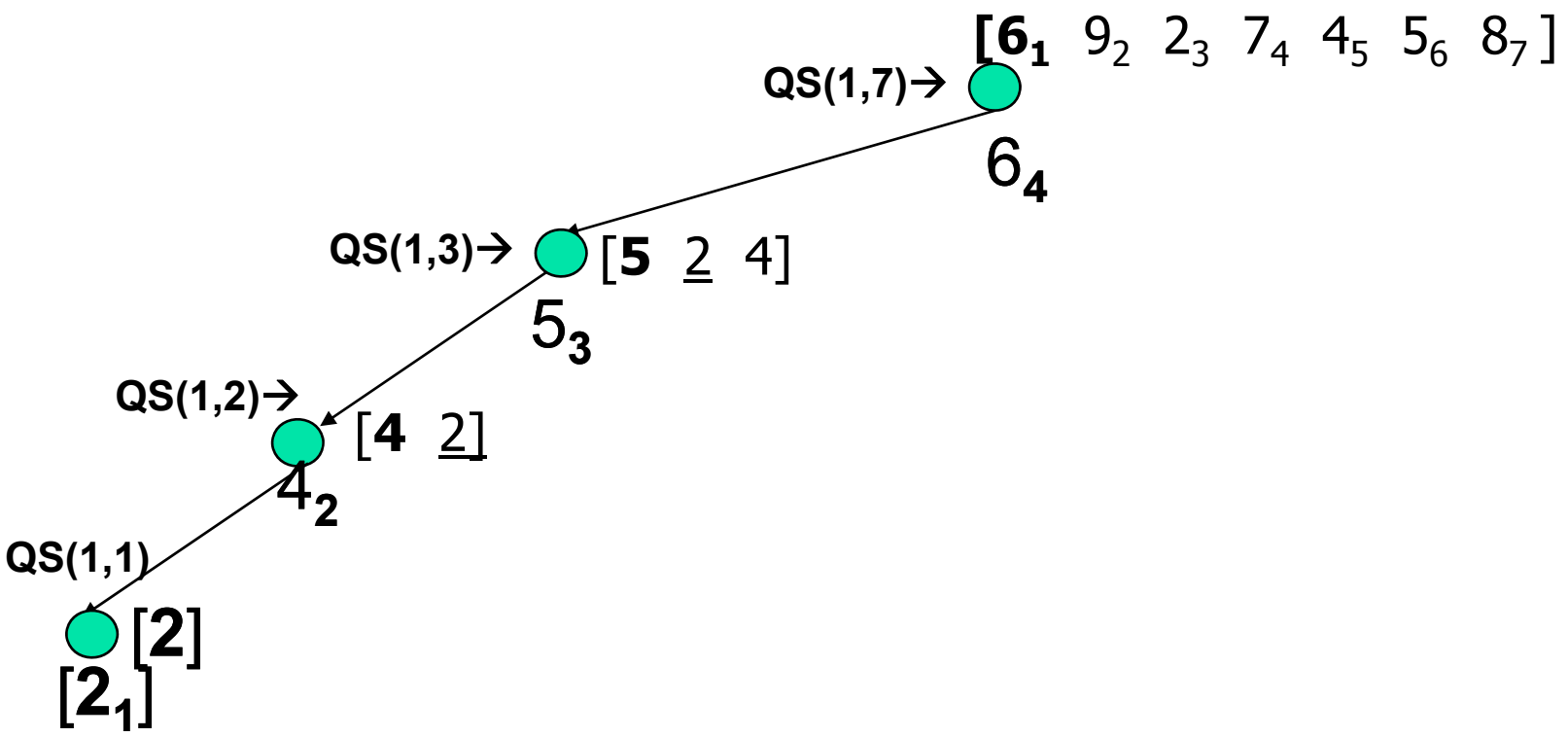
Quick Sort: αλγόριθμος

```
Quick Sort (A, f, r)
    if f < r then
        m=Partition (A, f, r)
        Quick Sort (A, f, m-1)
        Quick Sort (A, m+1 , r)
    end if
```

*****

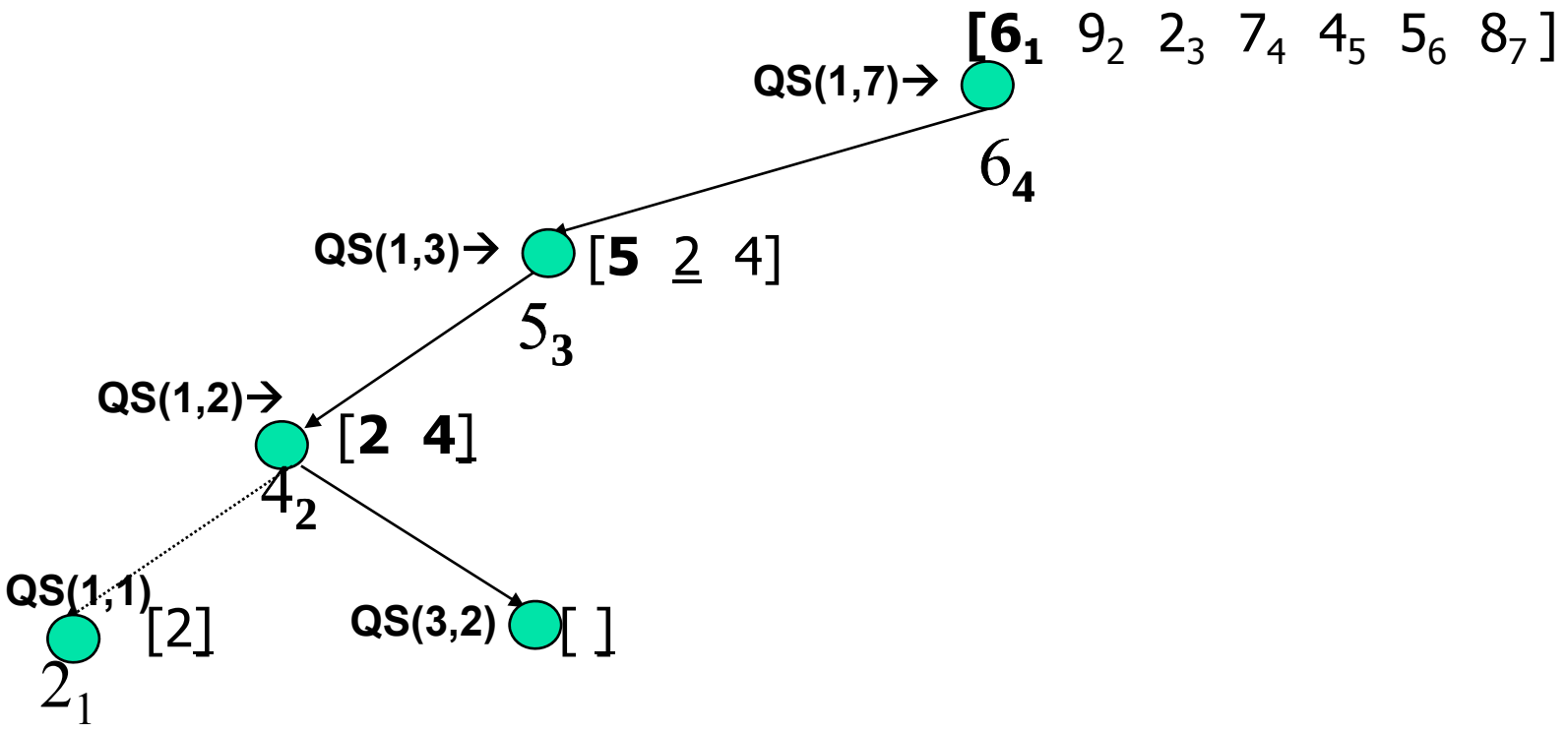
Quick Sort – αναδρομικό δέντρο

Παράδειγμα: $\mathbf{6_1}$ 9_2 2_3 7_4 4_5 5_6 8_7



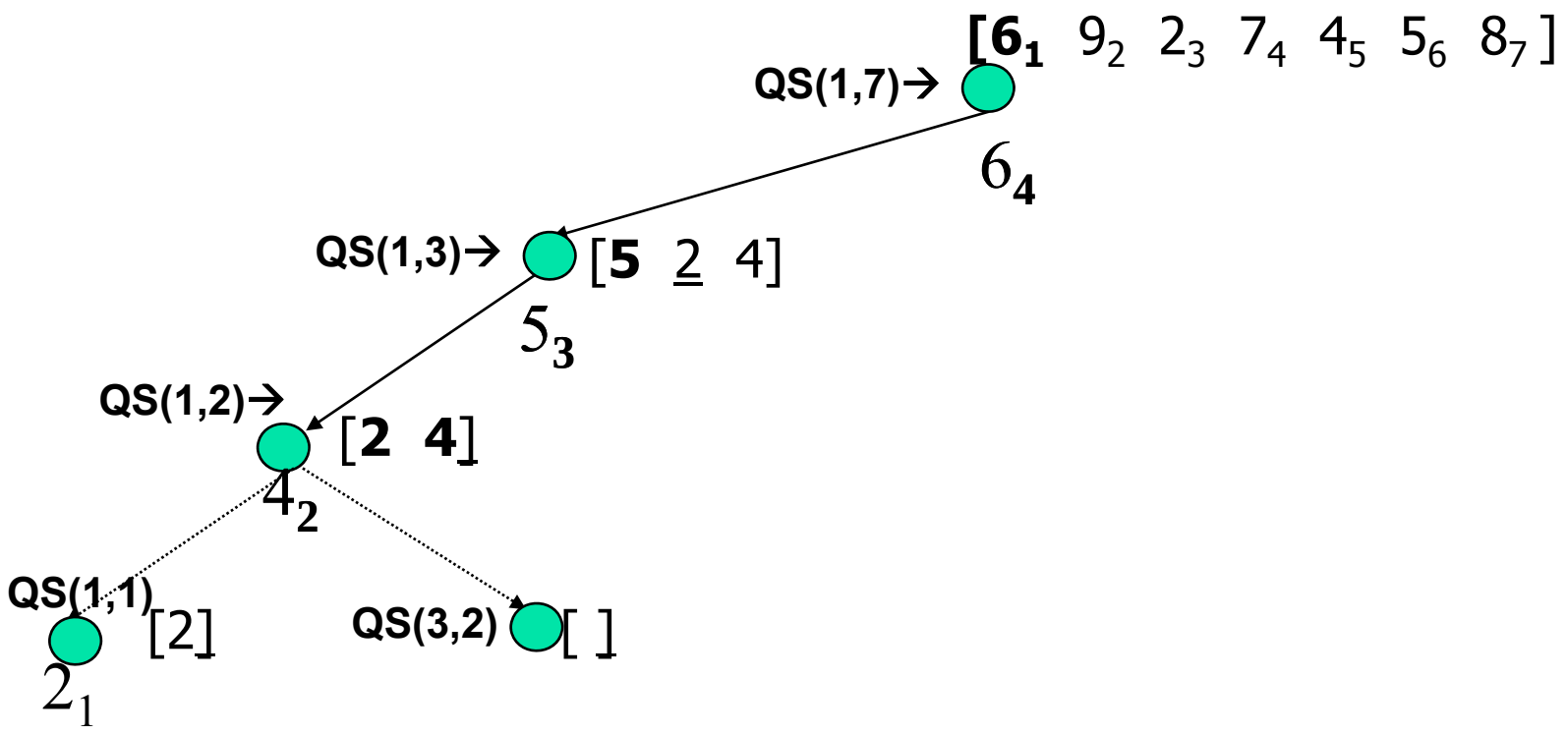
Quick Sort – αναδρομικό δέντρο

Παράδειγμα: $\mathbf{6_1}$ 9_2 2_3 7_4 4_5 5_6 8_7



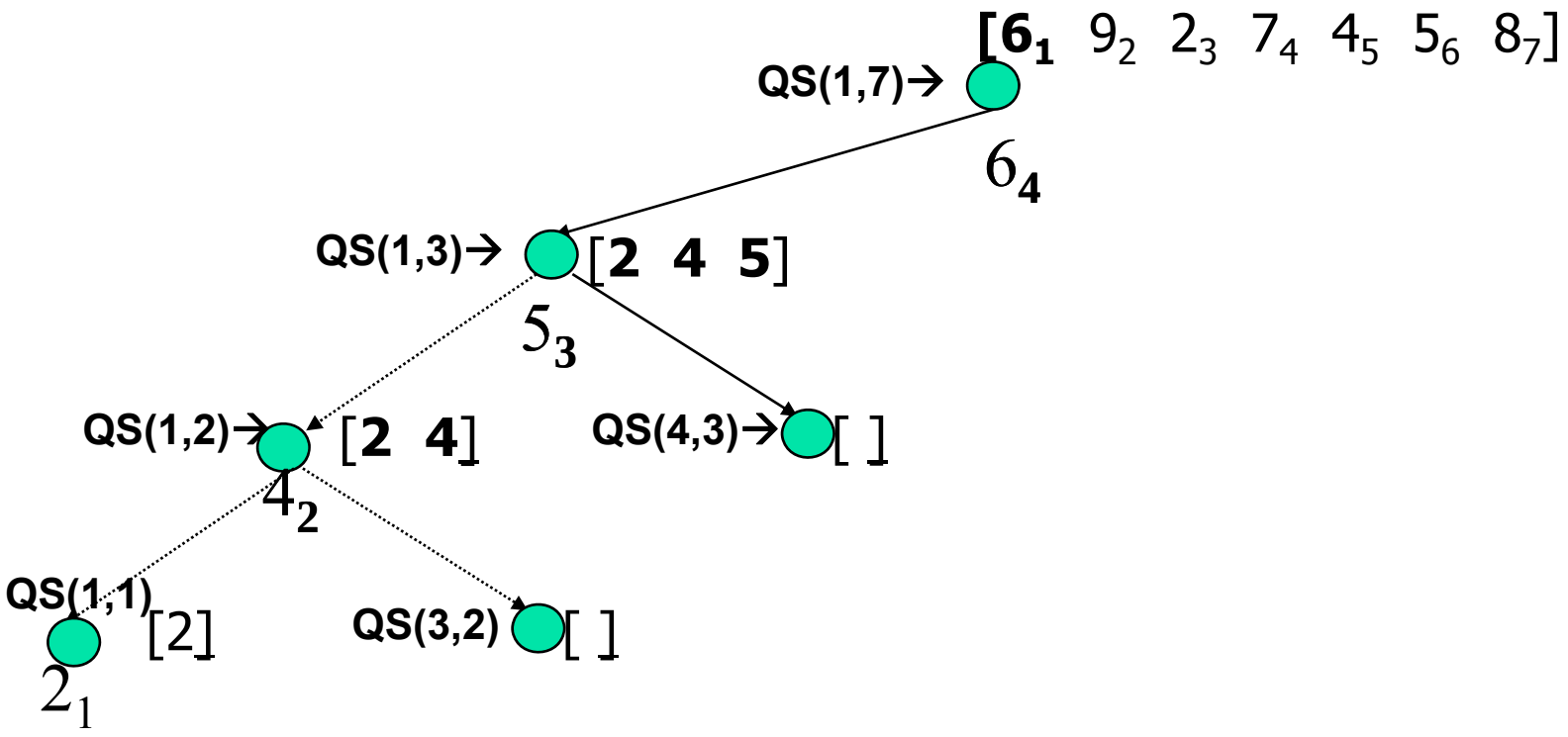
Quick Sort – αναδρομικό δέντρο

Παράδειγμα: $\mathbf{6_1}\ 9_2\ 2_3\ 7_4\ 4_5\ 5_6\ 8_7$



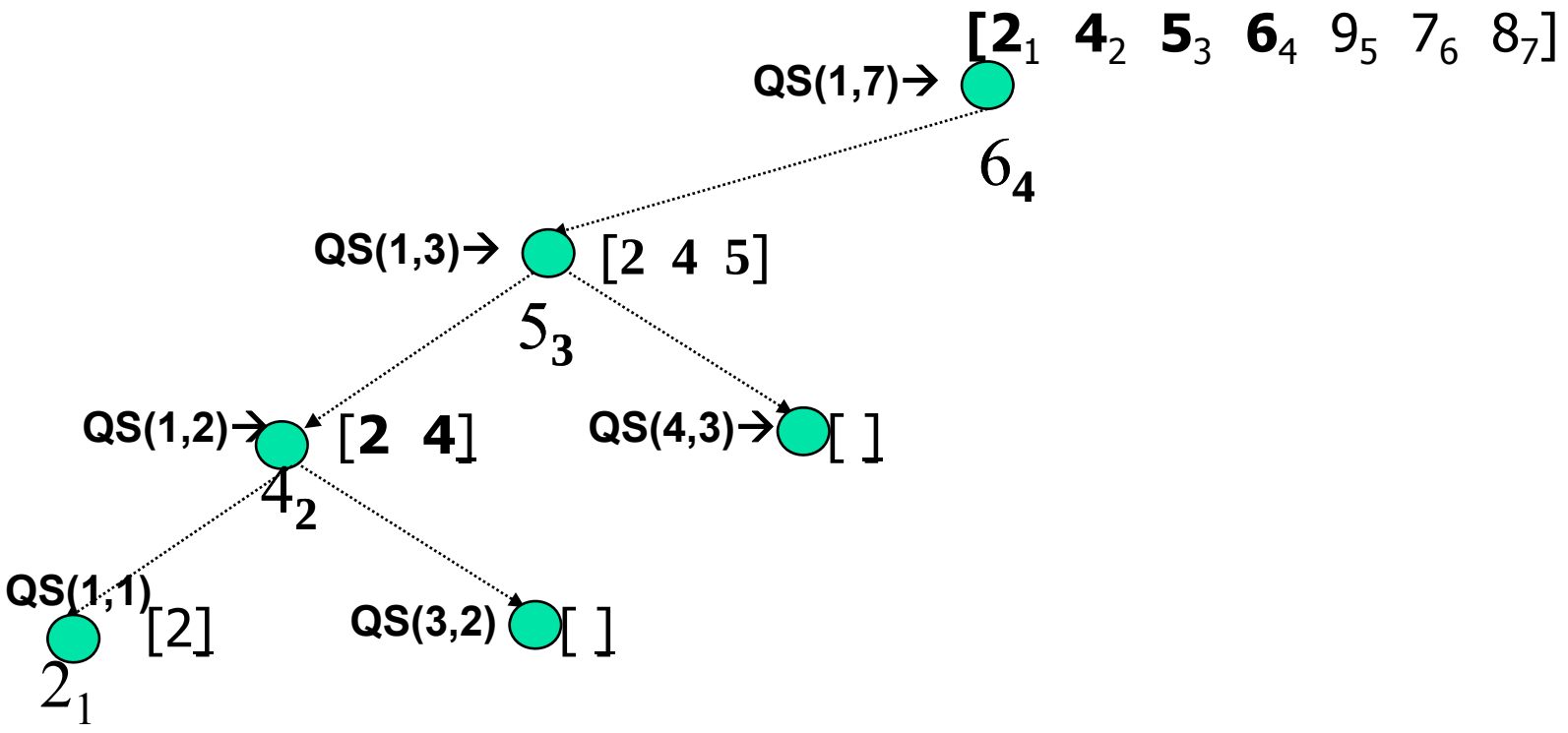
Quick Sort – αναδρομικό δέντρο

Παράδειγμα: $\mathbf{6_1}$ 9_2 2_3 7_4 4_5 5_6 8_7



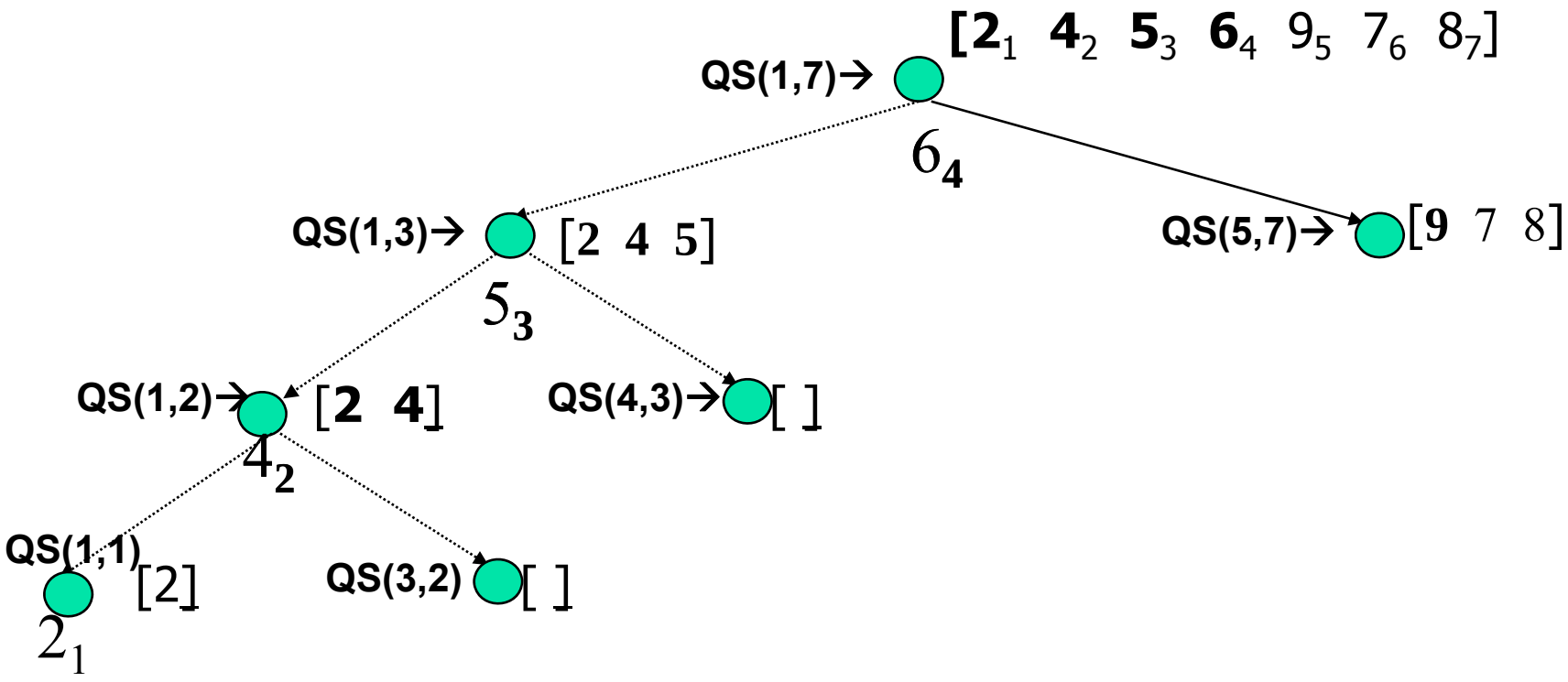
Quick Sort – αναδρομικό δέντρο

Παράδειγμα: $\mathbf{6_1}$ 9_2 2_3 7_4 4_5 5_6 8_7



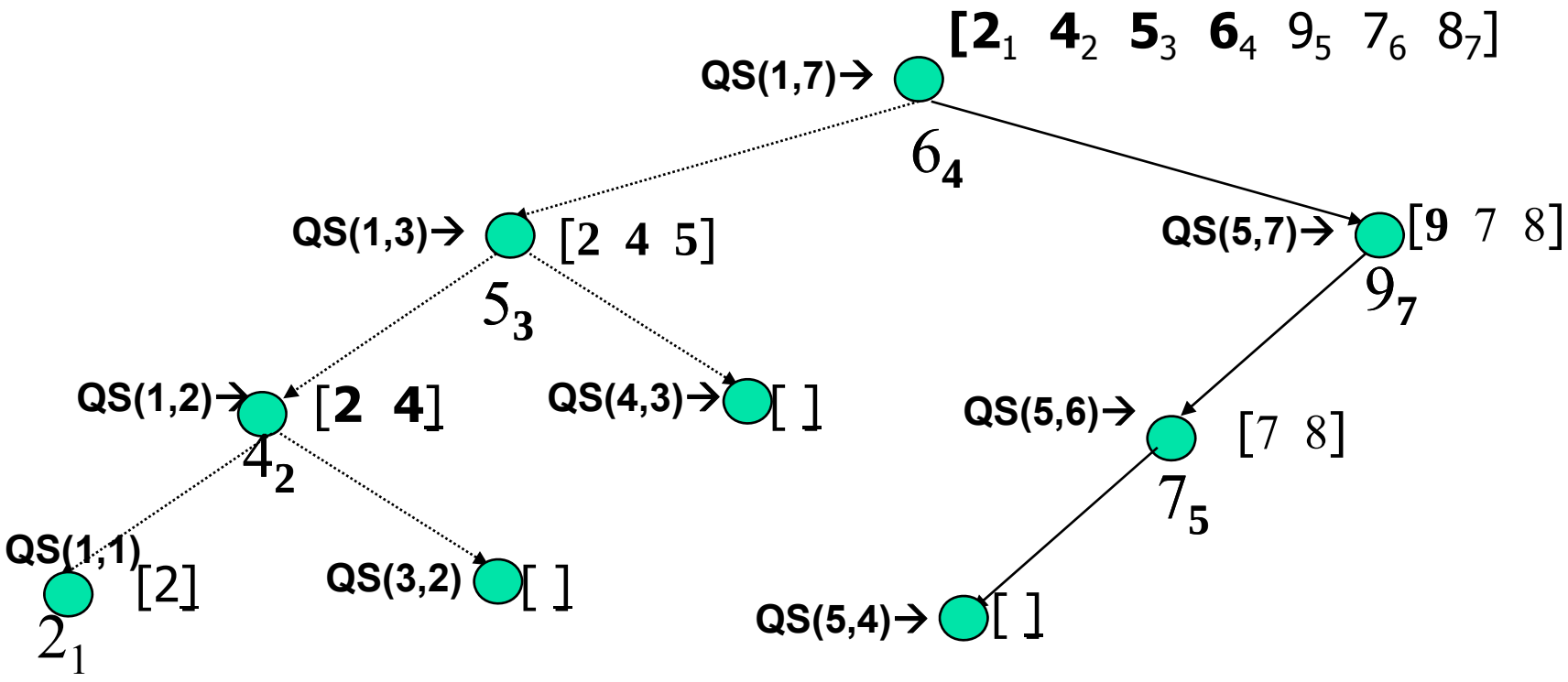
Quick Sort – αναδρομικό δέντρο

Παράδειγμα: $\mathbf{6_1}$ 9_2 2_3 7_4 4_5 5_6 8_7



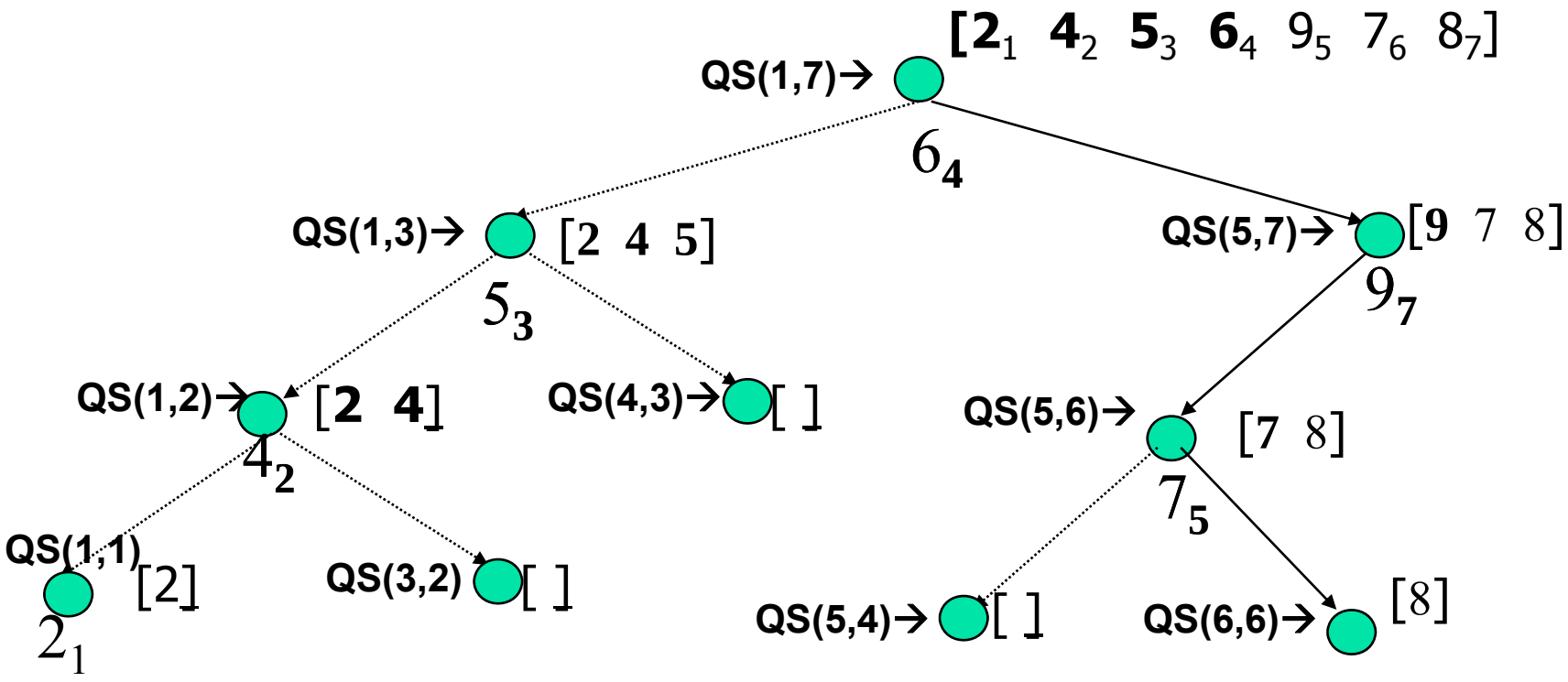
Quick Sort – αναδρομικό δέντρο

Παράδειγμα: $\mathbf{6_1}$ 9_2 2_3 7_4 4_5 5_6 8_7



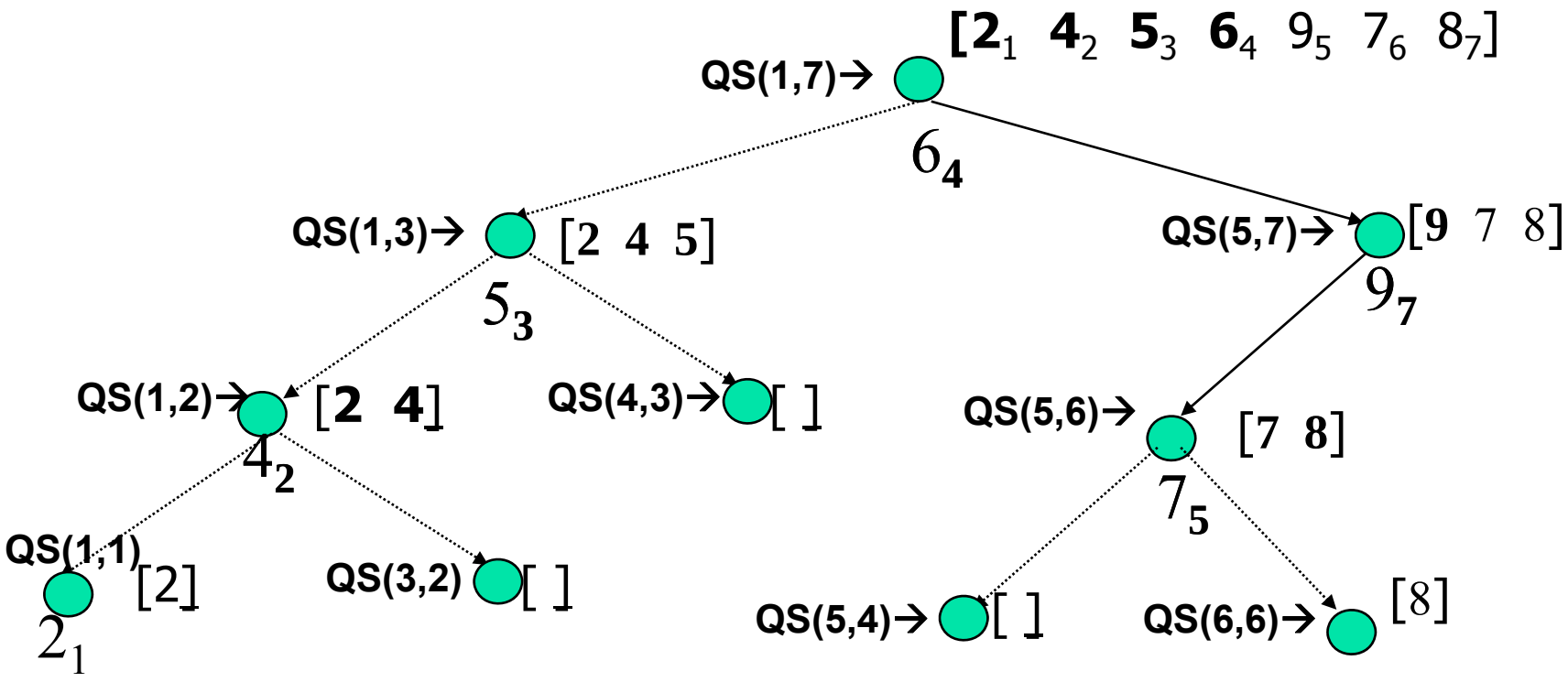
Quick Sort – αναδρομικό δέντρο

Παράδειγμα: **6_1** 9_2 2_3 7_4 4_5 5_6 8_7



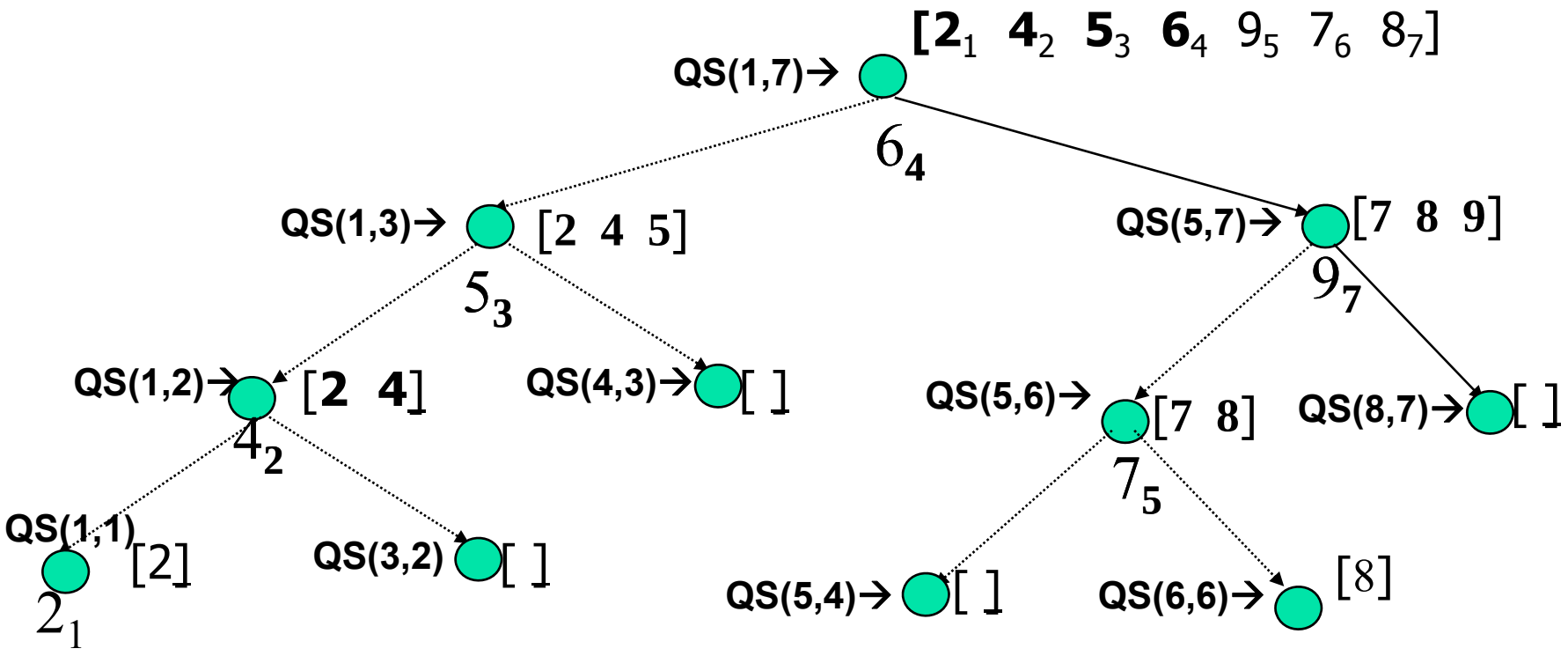
Quick Sort – αναδρομικό δέντρο

Παράδειγμα: $\mathbf{6_1}$ 9_2 2_3 7_4 4_5 5_6 8_7



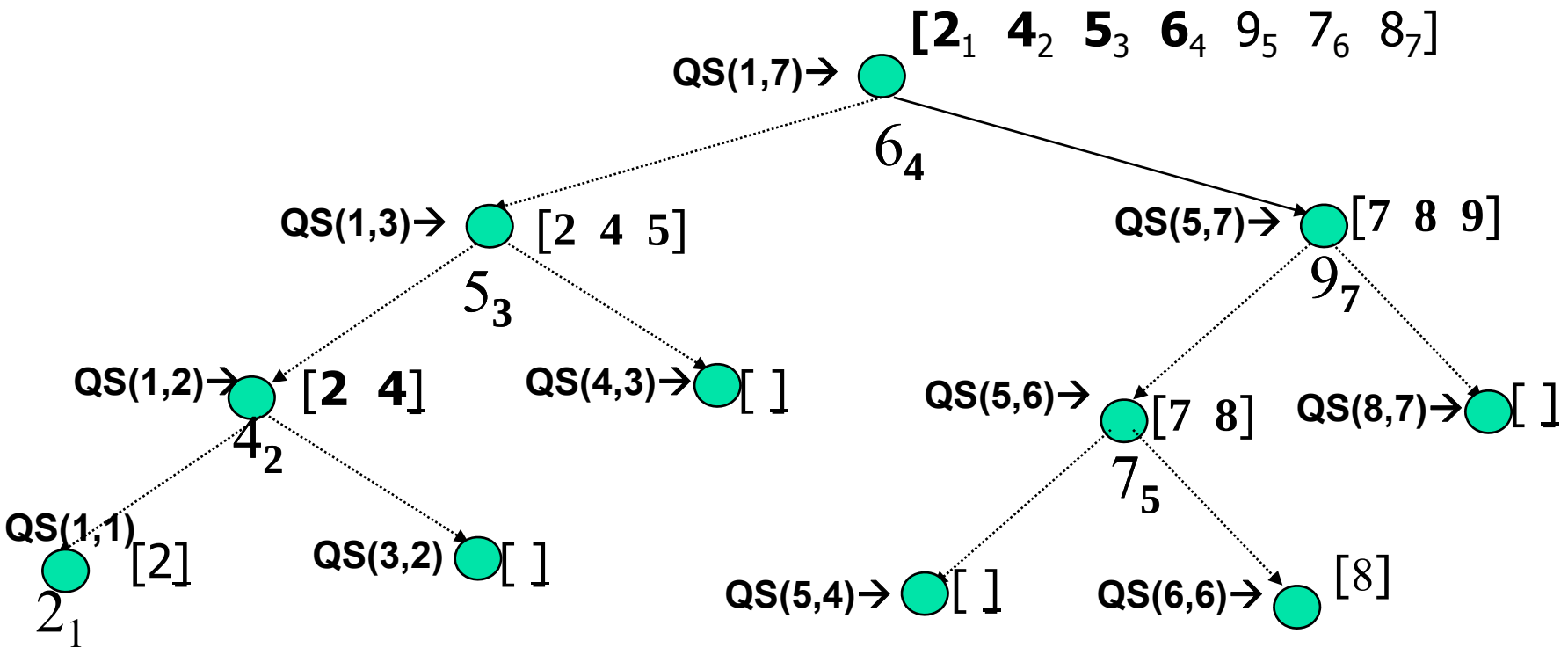
Quick Sort – αναδρομικό δέντρο

Παράδειγμα: $\mathbf{6_1}$ 9_2 2_3 7_4 4_5 5_6 8_7



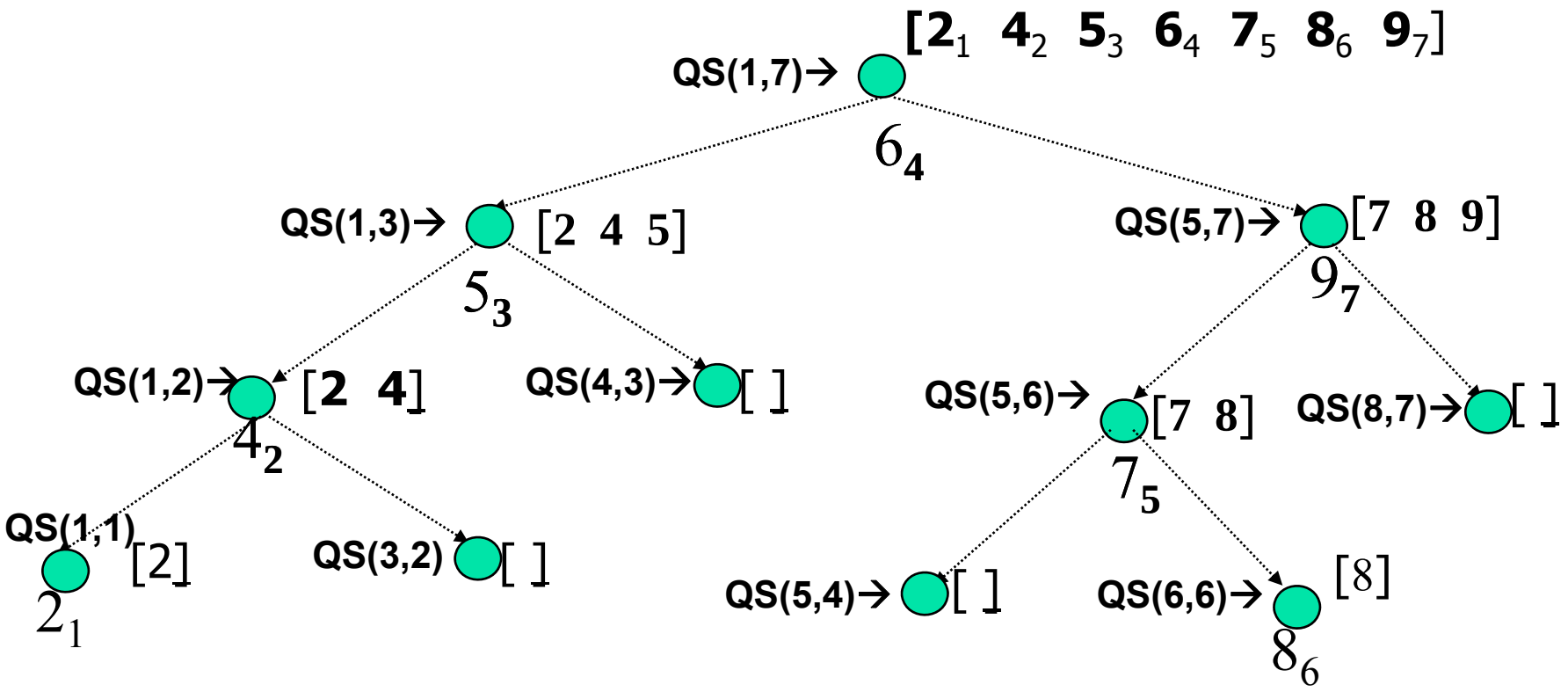
Quick Sort – αναδρομικό δέντρο

Παράδειγμα: $\mathbf{6_1}$ 9_2 2_3 7_4 4_5 5_6 8_7



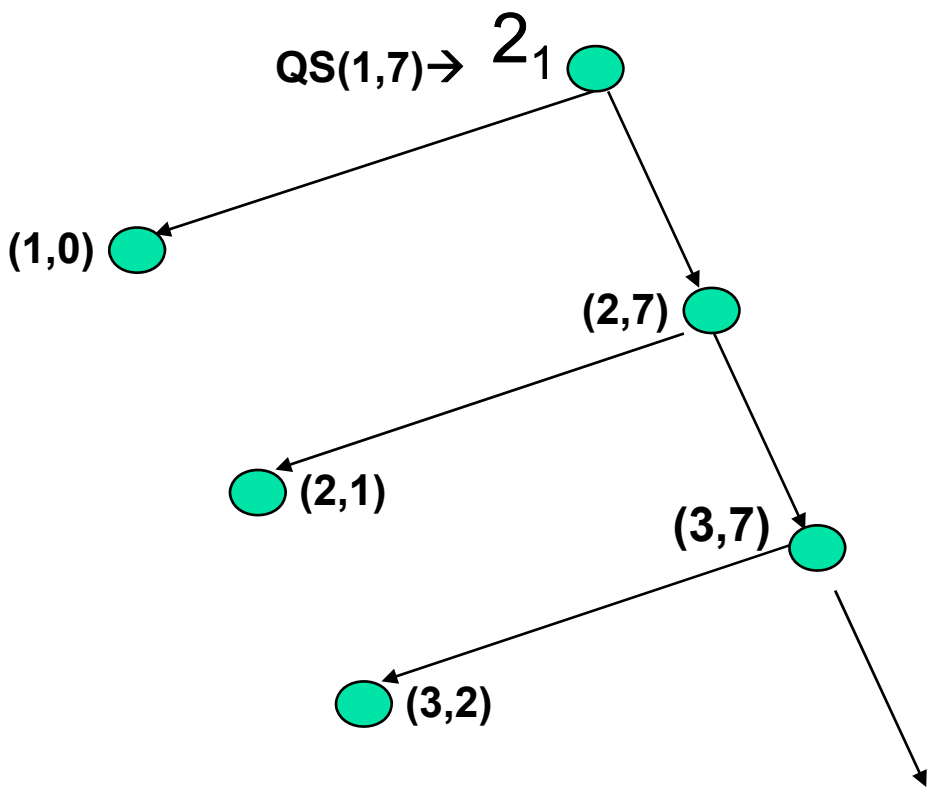
Quick Sort – αναδρομικό δέντρο

Παράδειγμα: $\mathbf{6_1}$ 9_2 2_3 7_4 4_5 5_6 8_7



Quick Sort - worst case

Παράδειγμα: 2_1 9_2 17_3 24_4 32_5 40_6 58_7



Πολυπλοκότητα Quick Sort

1. **Χείριστη περίπτωση**
   αύξουσες ή φθίνουσες ακολουθίες αριθμών
   →"διαμερίσεις εκφυλισμένες"

$$T_n = \begin{cases} n-1+T_{n-1} & \text{αν } n>1 \\ 1 & \text{διαφορετικά} \end{cases}$$

Επομένως $O(n^2)$

Πολυπλοκότητα Quick Sort

2. **Κατά μέσο όρο**
Όλες οι θέσεις για την οριστική θέση του στοιχείου θ είναι ισοπίθανες c= $\frac{1}{n}$

Έχουμε $T_n = n-1+ \frac{1}{n} \sum_{k=1}^{n}(T_{k-1}+ T_{n-k})$

και λόγω συμμετρίας $T_n = n-1+ \frac{2}{n} \sum_{k=1}^{n} T_{k-1}$

Τελικά $T_n = O(n\log n)$

Μέση πολυπλοκότητα Quick Sort

$T_0=T_1=0,\ n>1$

$T_n=n+1+\frac{1}{n}\sum_{k=1}^{n}(T_{k-1}+T_{n-k})$ και λόγω συμμετρίας

$T_n=n+1+\frac{2}{n}\sum_{k=1}^{n}T_{k-1} \rightarrow nT_n=n^2+n+2\sum_{k=1}^{n}T_{k-1}$

Μέση πολυπλοκότητα Quick Sort

Για n-1 έχουμε:

$$T_{n-1}=n+\frac{2}{n-1}\sum_{k=1}^{n-1}T_{k-1} \rightarrow (n-1)T_{n-1}=n^2-n+2\sum_{k=1}^{n-1}T_{k-1}$$

Αφαιρώντας κατά μέλη, έχουμε μετά την απλοποίηση:

$$nT_n=(n+1)T_{n-1}+2n$$

Μέση πολυπλοκότητα Quick Sort

Διαιρώντας με n(n+1) έχουμε:

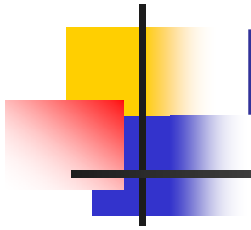
$$\frac{T_n}{n+1} = \frac{T_{n-1}}{n} + \frac{2}{n+1} = \frac{C_2}{3} + \sum_{k=3}^{n} \frac{2}{k+1}$$

Προσεγγίζοντας:

$$\frac{T_n}{n+1} \approx 2\sum_{k=1}^{n} \frac{1}{k} \approx 2\int_{1}^{n} \frac{1}{x}dx = 2\ln(n)$$

Τελικά: $T_n \approx 1{,}38n\log n$

Εναλλακτική λύση διαμέρισης



- Η προηγούμενη λύση δεν είναι συμμετρική

- Μια άλλη παρουσίαση του Quick Sort συνίσταται στο να εντοπίσουμε την τελική θέση του θ με δύο δείκτες που ξεκινούν από το 1 και το n και οι οποίοι συγκλίνουν προς την τελική θέση του θ.

- Χρησιμοποίηση "φρουρών" στα αριστερά και στα δεξιά του πίνακα. Μπορούμε να θέσουμε ένα μικρότερο στοιχείο από το θ στα αριστερά και ένα μεγαλύτερο στα δεξιά.