

Ταξινόμηση

Γιάννης Θεοδωρίδης, Νίκος Πελέκης, Άγγελος Πικράκης
Τμήμα Πληροφορικής

Ταξινόμηση

Έστω n διακριτές τιμές κλειδιών k_1, k_2, \dots, k_n και μια συλλογή T από n εγγραφές του τύπου

$$(k_1, I_1), (k_2, I_2), \dots, (k_n, I_n)$$

όπου I_j είναι η πληροφορία που σχετίζεται με το κλειδί k_j για $1 \leq j \leq n$.

Πρόβλημα ταξινόμησης: Να αναδιαταχτούν οι εγγραφές (k_j, I_j) στο T έτσι ώστε $k_i < k_{i+1}$, $1 \leq i < n$.

Μέτρα κόστους: Συγκρίσεις, Αντιμεταθέσεις

Θα δούμε 3 τεχνικές:

- Ευθείας εισαγωγής (Insertion sort)
- Ευθείας ανταλλαγής ή «Φυσαλίδας» (Bubble sort)
- Ευθείας επιλογής (Selection sort)

Ταξινόμηση ευθείας εισαγωγής (1)

Insertion sort

| | i=1 | 2 | 3 | 4 | 5 | 6 | 7 |
|----|-----|----|----|----|----|----|----|
| 42 | 20 | 17 | 13 | 13 | 13 | 13 | 13 |
| 20 | 42 | 20 | 17 | 17 | 14 | 14 | 14 |
| 17 | 17 | 42 | 20 | 20 | 17 | 17 | 15 |
| 13 | 13 | 13 | 42 | 28 | 20 | 20 | 17 |
| 28 | 28 | 28 | 28 | 42 | 28 | 23 | 20 |
| 14 | 14 | 14 | 14 | 14 | 42 | 28 | 23 |
| 23 | 23 | 23 | 23 | 23 | 23 | 42 | 28 |
| 15 | 15 | 15 | 15 | 15 | 15 | 15 | 42 |

Δομές Δεδομένων

3

Ταξινόμηση ευθείας εισαγωγής (2)

```
void Insertion_Sort(int A[], int n) {
    for (int i=1; i<n; i++)
        for (int j=i; (j>0) && (A[j] < A[j-1]); j--)
            swap(A, j, j-1);
}
```

Καλύτερη περίπτωση: 0 αντιμεταθέσεις, n-1 συγκρίσεις

Χειρότερη περίπτωση: $n^2/2$ αντιμεταθέσεις και συγκρίσεις

Μέση περίπτωση: $n^2/4$ αντιμεταθέσεις και συγκρίσεις

Δομές Δεδομένων

4

Ταξινόμηση ευθείας ανταλλαγής (1)

Bubble sort

| | i=0 | 1 | 2 | 3 | 4 | 5 | 6 |
|----|-----|----|----|----|----|----|----|
| 42 | 13 | 13 | 13 | 13 | 13 | 13 | 13 |
| 20 | 42 | 14 | 14 | 14 | 14 | 14 | 14 |
| 17 | 20 | 42 | 15 | 15 | 15 | 15 | 15 |
| 13 | 17 | 20 | 42 | 17 | 17 | 17 | 17 |
| 28 | 14 | 17 | 20 | 42 | 20 | 20 | 20 |
| 14 | 28 | 15 | 17 | 20 | 42 | 23 | 23 |
| 23 | 15 | 28 | 23 | 23 | 23 | 42 | 28 |
| 15 | 23 | 23 | 28 | 28 | 28 | 28 | 42 |

Δομές Δεδομένων

5

Ταξινόμηση ευθείας ανταλλαγής (2)

```
void Bubble_Sort(int A[], int n) {  
    for (int i=0; i<n-1; i++)  
        for (int j=n-1; j>i; j--)  
            if (A[j] < A[j-1])  
                swap(A, j, j-1);  
}
```

Καλύτερη περίπτωση: 0 αντιμεταθέσεις, $n^2/2$ συγκρίσεις

Χειρότερη περίπτωση: $n^2/2$ αντιμεταθέσεις και συγκρίσεις

Μέση περίπτωση: $n^2/4$ αντιμεταθέσεις, $n^2/2$ συγκρίσεις

Δομές Δεδομένων

6

Ταξινόμηση ευθείας επιλογής (1)

Selection sort

| | i=0 | 1 | 2 | 3 | 4 | 5 | 6 |
|----|-----|----|----|----|----|----|----|
| 42 | 13 | 13 | 13 | 13 | 13 | 13 | 13 |
| 20 | 20 | 14 | 14 | 14 | 14 | 14 | 14 |
| 17 | 17 | 17 | 15 | 15 | 15 | 15 | 15 |
| 13 | 42 | 42 | 42 | 17 | 17 | 17 | 17 |
| 28 | 28 | 28 | 28 | 28 | 20 | 20 | 20 |
| 14 | 14 | 20 | 20 | 20 | 28 | 23 | 23 |
| 23 | 23 | 23 | 23 | 23 | 23 | 28 | 28 |
| 15 | 15 | 15 | 17 | 42 | 42 | 42 | 42 |

Δομές Δεδομένων

7

Ταξινόμηση ευθείας επιλογής (2)

```
void Selection_Sort(int A[], int n) {
    for (int i=0; i<n-1; i++) {
        int lowindex = i; // Αρχικοποίηση ελαχίστου
        for (int j=n-1; j>i; j--)
            if (A[j] < A[lowindex])
                lowindex = j; // Εύρεση ελαχίστου
        swap(A, i, lowindex); // Τοποθέτηση στη σωστή θέση
    }
}
```

Καλύτερη περίπτωση: 0 (ή n-1) αντιμεταθέσεις, $n^2/2$ συγκρίσεις

Χειρότερη περίπτωση: n-1 αντιμεταθέσεις, $n^2/2$ συγκρίσεις

Μέση περίπτωση: $\Theta(n)$ αντιμεταθέσεις, $n^2/2$ συγκρίσεις

Δομές Δεδομένων

8

Συγκριτικά

| | Ευθείας Εισαγωγής | Ευθείας Ανταλλαγής | Ευθείας Επιλογής |
|------------------------|----------------------|-----------------------|---------------------|
| <u>Συγκρίσεις:</u> | | | |
| Καλύτερη περίπτωση | $O(n)$ | $O(n^2)$ | $O(n^2)$ |
| Μέση περίπτωση | $O(n^2)$ | $O(n^2)$ | $O(n^2)$ |
| Χειρότερη περίπτωση | $O(n^2)$ | $O(n^2)$ | $O(n^2)$ |
| <u>Αντιμεταθέσεις:</u> | | | |
| Καλύτερη περίπτωση | 0 | 0 | $O(n)$ |
| Μέση περίπτωση | $O(n^2)$ | $O(n^2)$ | $O(n)$ |
| Χειρότερη περίπτωση | $O(n^2)$ | $O(n^2)$ | $O(n)$ |