

Επανάληψη Ύλης

Εργαστήριο Αλγορίθμων

2020 - 2021

Εξεταστέα Ύλη:

Αλγόριθμοι Αναζήτησης

- Δυαδική Αναζήτηση (Binary Search) με επαναληπτική εντολή
- Αναζήτηση Παρεμβολής (Interpolation Search)

Επαναληπτικοί Αλγόριθμοι Ταξινόμησης

- Ταξινόμηση με Εισαγωγή (Insertion Sort)
- Ταξινόμηση με Επιλογή (Selection Sort)
- Ταξινόμηση με Αντιμετάθεση (Bubble Sort)

Αναδρομικοί Αλγόριθμοι Ταξινόμησης

- Ταξινόμηση Συγχώνευσης (Merge Sort)
- Γρήγορη Ταξινόμηση (Quick Sort)

Γράφοι

- Αποθήκευση Γράφου με χρήση Δυσδιάστατου Πίνακα Γειτονικών Κορυφών
- Αποθήκευση Γράφου με χρήση Λιστών Γειτονικών Κορυφών

BONUS

- BFS: https://www.tutorialspoint.com/data_structures_algorithms/breadth_first_traversal.htm
- DFS: https://www.tutorialspoint.com/data_structures_algorithms/depth_first_traversal.htm

Εξεταστέα Ύλη:

Αλγόριθμοι Αναζήτησης

- Δυαδική Αναζήτηση (Binary Search) με επαναληπτική εντολή
- Αναζήτηση Παρεμβολής (Interpolation Search)

Επαναληπτικοί Αλγόριθμοι Ταξινόμησης

- Ταξινόμηση με Εισαγωγή (Insertion Sort)
- Ταξινόμηση με Επιλογή (Selection Sort)
- Ταξινόμηση με Αντιμετάθεση (Bubble Sort)

Αναδρομικοί Αλγόριθμοι Ταξινόμησης

- Ταξινόμηση Συγχώνευσης (Merge Sort)
- Γρήγορη Ταξινόμηση (Quick Sort)

Γράφοι

- Αποθήκευση Γράφου με χρήση Δυσδιάστατου Πίνακα Γειτονικών Κορυφών
- Αποθήκευση Γράφου με χρήση Λιστών Γειτονικών Κορυφών

Δυαδική Αναζήτηση (Binary Search)

- Αλγόριθμος αναζήτησης, ο οποίος βρίσκει τη θέση του ζητούμενου στοιχείου σε ένα ταξινομημένο πίνακα.
- Ασυμπτωτική πολυπλοκότητα: $O(N)$.
- Λειτουργία:

Συγκρίνει το ζητούμενο στοιχείο με αυτό που βρίσκεται στην μεσαία θέση του πίνακα.

Εάν είναι ίδια, η αναζήτηση τελειώνει εφόσον το στοιχείο έχει βρεθεί.

Εάν το ζητούμενο στοιχείο είναι μεγαλύτερο από αυτό της μεσαίας θέσης, τότε η αναζήτηση συνεχίζεται στο δεξιό τμήμα του πίνακα.

Αλλιώς, η αναζήτηση συνεχίζεται στο αριστερό τμήμα του πίνακα.

Η σύγκριση αυτή επαναλαμβάνεται μέχρι να βρεθεί το ζητούμενο στοιχείο ή μέχρι να μην υπάρχουν άλλα στοιχεία για σύγκριση.

Π.χ. Πόσες επαναλήψεις θα εκτελεστούν μέχρι να βρεθεί ο αριθμός 30;

		Στοιχεία Πίνακα							
		0	1	2	3	4	5	6	7
Αριθμός Επανάληψης		5	6	7	9	10	20	30	40
1		5	6	7	9	10	20	30	40
2		5	6	7	9	10	20	30	40
3		5	6	7	9	10	20	30	40

Εξεταστέα Ύλη:

Αλγόριθμοι Αναζήτησης

- Δυαδική Αναζήτηση (Binary Search) με επαναληπτική εντολή
- Αναζήτηση Παρεμβολής (Interpolation Search)

Επαναληπτικοί Αλγόριθμοι Ταξινόμησης

- Ταξινόμηση με Εισαγωγή (Insertion Sort)
- Ταξινόμηση με Επιλογή (Selection Sort)
- Ταξινόμηση με Αντιμετάθεση (Bubble Sort)

Αναδρομικοί Αλγόριθμοι Ταξινόμησης

- Ταξινόμηση Συγχώνευσης (Merge Sort)
- Γρήγορη Ταξινόμηση (Quick Sort)

Γράφοι

- Αποθήκευση Γράφου με χρήση Δυσδιάστατου Πίνακα Γειτονικών Κορυφών
- Αποθήκευση Γράφου με χρήση Λιστών Γειτονικών Κορυφών

Αναζήτηση Παρεμβολής (Interpolation Search)

- Αλγόριθμος αναζήτησης, ο οποίος βρίσκει τη θέση του ζητούμενου στοιχείου σε ένα ταξινομημένο πίνακα.
- Ασυμπτωτική πολυπλοκότητα: $O(N)$.
- Λειτουργεί με όμοιο τρόπο όπως αυτόν της Δυαδικής Αναζήτησης, με την μόνη κύρια διαφορά να βρίσκεται στον τρόπο υπολογισμού της μεσαίας θέσης. Η συγκεκριμένη αναζήτηση χρησιμοποιεί έναν ειδικό τύπο για να βρει την θέση με την οποία θα συγκρίνει το ζητούμενο στοιχείο.
- Ο ειδικός τύπος προσπαθεί να βρει κατά προσέγγιση την πιο πιθανή θέση στην οποία μπορεί να βρίσκεται το ζητούμενο στοιχείο. Γι' αυτόν τον λόγο και ο συγκεκριμένος αλγόριθμος λειτουργεί πιο αποδοτικά εάν τα στοιχεία του πίνακα είναι ομοιόμορφα κατανεμημένα μεταξύ τους.

$$\text{mid} = \text{low} + \frac{(\text{high} - \text{low}) * (\text{num} - \text{pin}[\text{low}])}{\text{pin}[\text{high}] - \text{pin}[\text{low}]}$$

Π.χ. Πόσες επαναλήψεις θα εκτελεστούν μέχρι να βρεθεί ο αριθμός 20;

		Στοιχεία Πίνακα							
		0	1	2	3	4	5	6	7
Αριθμός Επανάληψης		5	6	7	9	10	20	30	40
1		5	6	7	9	10	20	30	40
2		5	6	7	9	10	20	30	40
3		5	6	7	9	10	20	30	40

Εξεταστέα Ύλη:

Αλγόριθμοι Αναζήτησης

- Δυαδική Αναζήτηση (Binary Search) με επαναληπτική εντολή
- Αναζήτηση Παρεμβολής (Interpolation Search)

Επαναληπτικοί Αλγόριθμοι Ταξινόμησης

- Ταξινόμηση με Εισαγωγή (Insertion Sort)
- Ταξινόμηση με Επιλογή (Selection Sort)
- Ταξινόμηση με Αντιμετάθεση (Bubble Sort)

Αναδρομικοί Αλγόριθμοι Ταξινόμησης

- Ταξινόμηση Συγχώνευσης (Merge Sort)
- Γρήγορη Ταξινόμηση (Quick Sort)

Γράφοι

- Αποθήκευση Γράφου με χρήση Δυσδιάστατου Πίνακα Γειτονικών Κορυφών
- Αποθήκευση Γράφου με χρήση Λιστών Γειτονικών Κορυφών

Ταξινόμηση με Εισαγωγή (Insertion Sort)

- Αλγόριθμος ταξινόμησης με χρήση επαναληπτικής εντολής.
- Ασυμπτωτική πολυπλοκότητα: $O(N^2)$.
- Λειτουργία:



Ο πίνακας χωρίζεται σε 2 τμήματα: το ταξινομημένο μέρος στα αριστερά και το αταξιινόμητο μέρος στα δεξιά.

Κατά την πρώτη εκτέλεση του αλγορίθμου, ως ταξινομημένο τμήμα θεωρείται η μηδενική θέση του πίνακα και ως αταξιινόμητο όλες οι υπόλοιπες θέσεις στα δεξιά.

Κάθε φορά επιλέγεται το πρώτο στοιχείο από το αταξιινόμητο μέρος και τοποθετείται στην σωστή θέση αριστερά. Αυτό επαναλαμβάνεται μέχρι να επιλεχθούν όλα τα στοιχεία για ταξινόμηση.

Π.χ. Πόσες επαναλήψεις (του εξωτερικού for) θα εκτελεστούν μέχρι την ταξινόμηση του παρακάτω πίνακα;

Στοιχεία Πίνακα					
	0	1	2	3	4
Αριθμός Επανάληψης	52	12	71	56	5
1	12	52	71	56	5
2	12	52	71	56	5
3	12	52	56	71	5
4	5	12	52	56	71

 αταξιινόμητο τμήμα
 ταξινομημένο τμήμα

Εξεταστέα Ύλη:

Αλγόριθμοι Αναζήτησης

- Δυαδική Αναζήτηση (Binary Search) με επαναληπτική εντολή
- Αναζήτηση Παρεμβολής (Interpolation Search)

Επαναληπτικοί Αλγόριθμοι Ταξινόμησης

- Ταξινόμηση με Εισαγωγή (Insertion Sort)
- Ταξινόμηση με Επιλογή (Selection Sort)
- Ταξινόμηση με Αντιμετάθεση (Bubble Sort)

Αναδρομικοί Αλγόριθμοι Ταξινόμησης

- Ταξινόμηση Συγχώνευσης (Merge Sort)
- Γρήγορη Ταξινόμηση (Quick Sort)

Γράφοι

- Αποθήκευση Γράφου με χρήση Δυσδιάστατου Πίνακα Γειτονικών Κορυφών
- Αποθήκευση Γράφου με χρήση Λιστών Γειτονικών Κορυφών

Ταξινόμηση με Επιλογή (Selection Sort)

- Αλγόριθμος ταξινόμησης με χρήση επαναληπτικής εντολής.
- Ασυμπτωτική πολυπλοκότητα: $O(N^2)$.
- Λειτουργία:



Ο πίνακας χωρίζεται σε 2 τμήματα: το ταξινομημένο μέρος στα αριστερά και το αταξιινόμητο μέρος στα δεξιά.

Κατά την πρώτη εκτέλεση του αλγορίθμου, το ταξινομημένο τμήμα είναι κενό και όλος ο πίνακας θεωρείται αταξινόμητος.

Κάθε φορά επιλέγεται το ελάχιστο στοιχείο από το αταξινόμητο μέρος και ανταλλάσσεται με το πρώτο αταξινόμητο. Αυτό επαναλαμβάνεται μέχρι να επιλεγθούν όλα τα στοιχεία για ταξινόμηση.

Π.χ. Πόσες επαναλήψεις (του εξωτερικού for) θα εκτελεστούν μέχρι την ταξινόμηση του παρακάτω πίνακα;

Στοιχεία Πίνακα					
	0	1	2	3	4
Αριθμός Επανάληψης	52	12	71	56	5
1	5	12	71	56	52
2	5	12	71	56	52
3	5	12	52	56	71
4	5	12	52	56	71
5	5	12	52	56	71

 αταξινόμητο τμήμα
 ταξινομημένο τμήμα

Εξεταστέα Ύλη:

Αλγόριθμοι Αναζήτησης

- Δυαδική Αναζήτηση (Binary Search) με επαναληπτική εντολή
- Αναζήτηση Παρεμβολής (Interpolation Search)

Επαναληπτικοί Αλγόριθμοι Ταξινόμησης

- Ταξινόμηση με Εισαγωγή (Insertion Sort)
- Ταξινόμηση με Επιλογή (Selection Sort)
- Ταξινόμηση με Αντιμετάθεση (Bubble Sort)

Αναδρομικοί Αλγόριθμοι Ταξινόμησης

- Ταξινόμηση Συγχώνευσης (Merge Sort)
- Γρήγορη Ταξινόμηση (Quick Sort)

Γράφοι

- Αποθήκευση Γράφου με χρήση Δυσδιάστατου Πίνακα Γειτονικών Κορυφών
- Αποθήκευση Γράφου με χρήση Λιστών Γειτονικών Κορυφών

Ταξινόμηση με Αντιμετάθεση (Bubble Sort)

- Αλγόριθμος ταξινόμησης με χρήση επαναληπτικής εντολής.
- Ασυμπτωτική πολυπλοκότητα: $O(N^2)$.
- Λειτουργία:



Ο πίνακας χωρίζεται σε 2 τμήματα: το ταξινομημένο μέρος στα αριστερά και το αταξιινόμητο μέρος στα δεξιά.

Κατά την πρώτη εκτέλεση του αλγορίθμου, το ταξινομημένο τμήμα είναι κενό και όλος ο πίνακας θεωρείται αταξινόμητος.

Κάθε φορά ξεκινώντας από το τέλος του αταξινόμητου μέρους, συγκρίνονται τα δύο γειτονικά στοιχεία και εναλλάσσονται εάν το μεγαλύτερο βρίσκεται αριστερά από το μικρότερο. Στην συνέχεια επιλέγονται τα επόμενα γειτονικά μέχρι την αρχή του αταξινόμητου πίνακα. Αυτό επαναλαμβάνεται $N-1$ φορές, όπου N το μέγεθος του πίνακα.

Π.χ. Πόσες επαναλήψεις (του εξωτερικού for) θα εκτελεστούν μέχρι την ταξινόμηση του παρακάτω πίνακα;

		Στοιχεία Πίνακα				
		0	1	2	3	4
Αριθμός Επανάληψης		52	12	71	56	5
1		5	52	12	71	56
2		5	12	52	56	71
3		5	12	52	56	71
4		5	12	52	56	71

 αταξιινόμητο τμήμα
 ταξινομημένο τμήμα

Εξεταστέα Ύλη:

Αλγόριθμοι Αναζήτησης

- Δυαδική Αναζήτηση (Binary Search) με επαναληπτική εντολή
- Αναζήτηση Παρεμβολής (Interpolation Search)

Επαναληπτικοί Αλγόριθμοι Ταξινόμησης

- Ταξινόμηση με Εισαγωγή (Insertion Sort)
- Ταξινόμηση με Επιλογή (Selection Sort)
- Ταξινόμηση με Αντιμετάθεση (Bubble Sort)

Αναδρομικοί Αλγόριθμοι Ταξινόμησης

- Ταξινόμηση Συγχώνευσης (Merge Sort)
- Γρήγορη Ταξινόμηση (Quick Sort)

Γράφοι

- Αποθήκευση Γράφου με χρήση Δυσδιάστατου Πίνακα Γειτονικών Κορυφών
- Αποθήκευση Γράφου με χρήση Λιστών Γειτονικών Κορυφών

Ταξινόμηση Συγχώνευσης (Merge Sort)

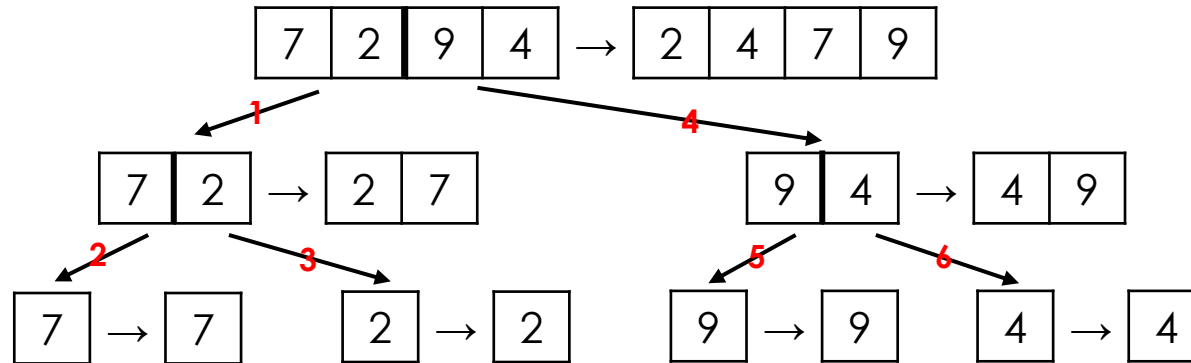
- Ακολουθεί τη λογική «Διαίρει – και – Βασίλευε», συνεπώς είναι αναδρομικός αλγόριθμος.
- Ασυμπτωτική πολυπλοκότητα: $O(N \log N)$.
- Λειτουργία:

Ο πίνακας χωρίζεται σε 2 ίσα τμήματα: το αριστερό και το δεξί.

Ταξινομείται αναδρομικά το αριστερό και το δεξί τμήμα.

Τα δύο ταξινομημένα τμήματα συγχωνεύονται σε ένα ενιαίο ταξινομημένο τμήμα.

Π.χ. Πόσες αναδρομικές κλήσεις θα εκτελεστούν (χωρίς την κύρια κλήση) μέχρι την ταξινόμηση του παρακάτω πίνακα;



Εξεταστέα Ύλη:

Αλγόριθμοι Αναζήτησης

- Δυαδική Αναζήτηση (Binary Search) με επαναληπτική εντολή
- Αναζήτηση Παρεμβολής (Interpolation Search)

Επαναληπτικοί Αλγόριθμοι Ταξινόμησης

- Ταξινόμηση με Εισαγωγή (Insertion Sort)
- Ταξινόμηση με Επιλογή (Selection Sort)
- Ταξινόμηση με Αντιμετάθεση (Bubble Sort)

Αναδρομικοί Αλγόριθμοι Ταξινόμησης

- Ταξινόμηση Συγχώνευσης (Merge Sort)
- Γρήγορη Ταξινόμηση (Quick Sort)

Γράφοι

- Αποθήκευση Γράφου με χρήση Δυσδιάστατου Πίνακα Γειτονικών Κορυφών
- Αποθήκευση Γράφου με χρήση Λιστών Γειτονικών Κορυφών

Γρήγορη Ταξινόμηση (Quick Sort)

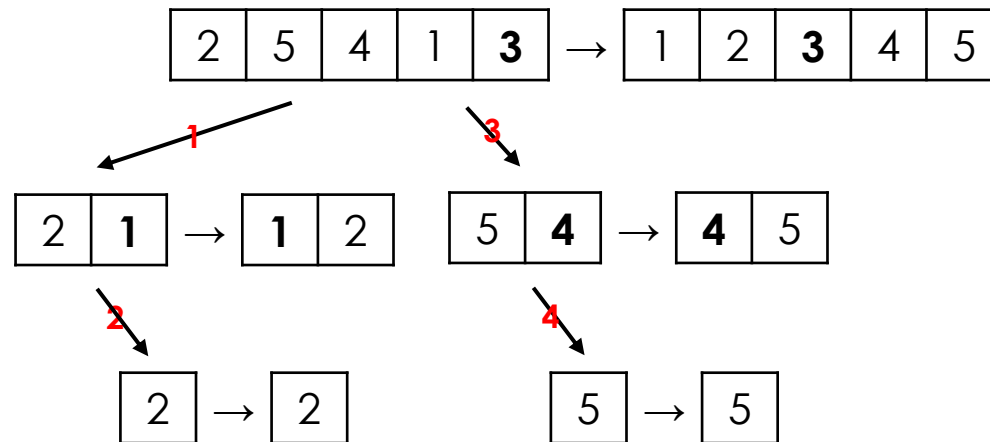
- Ακολουθεί τη λογική «Διαίρει – και – Βασίλευε», συνεπώς είναι αναδρομικός αλγόριθμος.
- Ασυμπτωτική πολυπλοκότητα: $O(N \log N)$.
- Λειτουργία:

Με βάση ένα τυχαίο στοιχείο ρινότ, χωρίζει τον πίνακα, έτσι ώστε τα στοιχεία μικρότερα του ρινότ βρίσκονται στα αριστερά του και τα μεγαλύτερα βρίσκονται στα δεξιά του.

Στην συνέχεια ταξινομείται αναδρομικά το αριστερό και δεξί τμήμα.

Ως ρινότ επιλέγεται το τελευταίο στοιχείο του εκάστοτε τμήματος.

Π.χ. Πόσες αναδρομικές κλήσεις θα εκτελεστούν (χωρίς την κύρια κλήση) μέχρι την ταξινόμηση του παρακάτω πίνακα;



Εξεταστέα Ύλη:

Αλγόριθμοι Αναζήτησης

- Δυαδική Αναζήτηση (Binary Search) με επαναληπτική εντολή
- Αναζήτηση Παρεμβολής (Interpolation Search)

Επαναληπτικοί Αλγόριθμοι Ταξινόμησης

- Ταξινόμηση με Εισαγωγή (Insertion Sort)
- Ταξινόμηση με Επιλογή (Selection Sort)
- Ταξινόμηση με Αντιμετάθεση (Bubble Sort)

Αναδρομικοί Αλγόριθμοι Ταξινόμησης

- Ταξινόμηση Συγχώνευσης (Merge Sort)
- Γρήγορη Ταξινόμηση (Quick Sort)

Γράφοι

- Αποθήκευση Γράφου με χρήση Δυσδιάστατου Πίνακα Γειτονικών Κορυφών
- Αποθήκευση Γράφου με χρήση Λιστών Γειτονικών Κορυφών

Αποθήκευση Γράφου με χρήση Δυσδιάστατου Πίνακα Γειτονικών Κορυφών

- Για την αναπαράσταση του γράφου χρησιμοποιείται τετραγωνικός πίνακας $N \times N$ (όπου N ο αριθμός των κορυφών).
- $\text{pin}[i][j] = 1$ εάν υπάρχει ακμή από την κορυφή i στην κορυφή j , αλλιώς $\text{pin}[i][j] = 0$.
- Στην περίπτωση μη – κατευθυνόμενου γράφου εάν υπάρχει ακμή μεταξύ των κορυφών i και j , τότε ισχύει $\text{pin}[i][j] = 1$ και $\text{pin}[j][i] = 1$.
- Το άθροισμα κάθε γραμμής δείχνει πόσες γειτονικές κορυφές έχει η κορυφή i (βαθμός κορυφής).
- Το άθροισμα κάθε στήλης δείχνει σε πόσες κορυφές είναι γειτονική η κορυφή i .

Π.χ. Για τον μη – κατευθυνόμενο γράφο με κορυφές 0, 1, 2, 3, 4 και ζευγάρια ακμών (0 – 1), (1 – 2), (1 – 3), (1 – 4), (2 – 4), (3 – 4) ποια θα είναι η αναπαράσταση του δυσδιάστατου πίνακα;

	0	1	2	3	4
0	0	1	0	0	0
1	1	0	1	1	1
2	0	1	0	0	1
3	0	1	0	0	1
4	0	1	1	1	0

Εξεταστέα Ύλη:

Αλγόριθμοι Αναζήτησης

- Δυαδική Αναζήτηση (Binary Search) με επαναληπτική εντολή
- Αναζήτηση Παρεμβολής (Interpolation Search)

Επαναληπτικοί Αλγόριθμοι Ταξινόμησης

- Ταξινόμηση με Εισαγωγή (Insertion Sort)
- Ταξινόμηση με Επιλογή (Selection Sort)
- Ταξινόμηση με Αντιμετάθεση (Bubble Sort)

Αναδρομικοί Αλγόριθμοι Ταξινόμησης

- Ταξινόμηση Συγχώνευσης (Merge Sort)
- Γρήγορη Ταξινόμηση (Quick Sort)

Γράφοι

- Αποθήκευση Γράφου με χρήση Δυσδιάστατου Πίνακα Γειτονικών Κορυφών
- Αποθήκευση Γράφου με χρήση Λιστών Γειτονικών Κορυφών

Αποθήκευση Γράφου με χρήση Λιστών Γειτονικών Κορυφών

- Για την αναπαράσταση του γράφου χρησιμοποιείται ένας πίνακα μεγέθους N (όπου N ο αριθμός των κορυφών), τα στοιχεία του οποίου είναι δομές. Κάθε δομή παριστάνει την κορυφή και περιέχει το όνομά της και ένα δείκτη που δείχνει σε μία απλά συνδεδεμένη λίστα με τις γειτονικές της κορυφές.
- Εναλλακτική υλοποίηση μπορεί να περιλαμβάνεις τον πίνακα μεγέθους N ως ένα πίνακα δεικτών, όπου κάθε δείκτης δείχνει σε μία απλά συνδεδεμένη λίστα με τις γειτονικές κορυφές.

Π.χ. Για τον μη – κατευθυνόμενο γράφο με κορυφές 0, 1, 2, 3, 4 και ζευγάρια ακμών (0 – 1), (1 – 2), (1 – 3), (1 – 4), (2 – 4), (3 – 4) ποια θα είναι η αναπαράσταση των λιστών με τις γειτονικές κορυφές;

0: 1
1: 0, 2, 3, 4
2: 1, 4
3: 1, 4
4: 1, 2, 3