

Αναφορά

Κωνσταντίνος Ιωάννης Κορναράκης p3170074

March 13, 2019

1 Πρώτη άσκηση

Εφόσον ο πίνακας είναι ταξινομημένος και θέλουμε να κάνουμε αναζήτηση σε $O(\log n)$, τότε είναι πιο κατάλληλο να χρησιμοποιήσουμε ως βάση τον αλγόριθμο της δυαδικής αναζήτησης. Δημιουργούμε δύο αλγορίθμους δυαδικής αναζήτησης με ορισμένες διαφοροποιήσεις έτσι ώστε ο ένας να μπορεί να εντοπίζει μόνο τη πρώτη εμφάνιση ενός κλειδιού, ενώ ο δεύτερος να μπορεί να εντοπίζει μόνο τη τελευταία εμφάνιση ενός κλειδιού.

Η `BinarySearchFirst` αν δει ότι η θέση του πίνακα που ελέγχουμε είναι η 0 ή στη θέση που ελέγχουμε το στοιχείο της ισούται με το κλειδί και το στοιχείο της προηγούμενης θέσης είναι μικρότερο του κλειδιού (άρα και όλα τα προηγούμενα στοιχεία είναι μικρότερα), τότε η αναζήτηση είναι επιτυχής και επιστρέφουμε. Αλλιώς, καλείται αναδρομικά στον υποπίνακα αριστερά ή δεξιά της θέσης της οποίας ελέγχουμε ανάλογα.

Παρόμοια η `BinarySearchLast`, με την διαφορά ότι αν στη θέση που ελέγχουμε το στοιχείο είναι ίσο με το κλειδί και στην επόμενη θέση το στοιχείο είναι μεγαλύτερο του κλειδιού τότε η αναζήτηση είναι επιτυχής.

$$T(N) = T\left(\frac{N}{2}\right) + O(1) \text{ για την χειρότερη/μέση περίπτωση}$$

Χειρότερη περίπτωση: $O(\log n)$

Μέση περίπτωση: $O(\log n)$

Καλύτερη περίπτωση: $O(1)$

Πολυπλοκότητα χώρου: $O(1)$

2 Δεύτερη άσκηση

α) Παράγουμε έναν ψευδοτυχαίο αριθμό μεταξύ του αριθμού της πρώτης θέσης του πίνακα και του αριθμού της τελικής θέσης του πίνακα. Κάνουμε αντιμετάθεση του στοιχείου που βρίσκεται στην τελευταία θέση του πίνακα με το στοιχείο που βρίσκεται στην θέση με αριθμό τον ψευδοτυχαίο αριθμό που παράχθηκε.

β) Δημιουργούμε τρεις μεταβλητές: έναν δείκτη ο οποίος δείχνει τη θέση του στοιχείου που εξετάζεται, ένα δείκτη της θέσης του πρώτου στοιχείου που είναι μεγαλύτερο από το ρινότ που αρχικά είναι ίσος με την πρώτη θέση του πίνακα, και έναν δείκτη της θέσης του τελευταίου στοιχείου που είναι μικρότερο από το ρινότ που αρχικά είναι ίσος με την τελευταία θέση του πίνακα. Διατρέχουμε τον πίνακα από την αρχή μέχρι το τέλος σε κάθε βήμα. Έχουμε ορίσει ως ρινότ το τελευταίο στοιχείο του πίνακα. Σε κάθε βήμα, κάνουμε ελέγχους: Εάν το στοιχείο το οποίο εξετάζεται είναι μικρότερο του ρινότ τότε, γίνεται αντιμετάθεση αυτού με το στοιχείο στη θέση του δείκτη αρχής, ο δείκτης αρχής αυξάνεται κατά ένα, όπως και ο δείκτης στοιχείου κατά 1. Εάν το στοιχείο το οποίο εξετάζεται είναι μεγαλύτερο από το ρινότ τότε, γίνεται αντιμετάθεση αυτού με το στοιχείο στην θέση του μετρητή τέλους, ο μετρητής τέλους μειώνεται κατά 1. Αλλιώς, αυξάνουμε τον δείκτη στοιχείου. Έτσι καταφέρνουμε να έχουμε σε κάθε αναδρομική κλήση όλα τα στοιχεία τα οποία είναι μικρότερα του ρινότ από την αριστερή μεριά του πίνακα (πριν τον δείκτη αρχής), όλα όσα είναι ίσα με το ρινότ στο κέντρο (μεταξύ του δείκτη αρχής και τέλους) και τα μεγαλύτερα από το ρινότ στη δεξιά μεριά (μετά τον δείκτη τέλους).

$T(n) = 3T(\frac{n}{3}) + f(n)$ (με $f(n) = cn$, για τον εντοπισμό του στοιχείου και την αντιμετάθεση στον πίνακα) για την μέση περίπτωση

Χειρότερη περίπτωση: $O(n^2)$ όταν κάνει n^2 συγκρίσεις

Μέση περίπτωση: $O(n \log n)$

Καλύτερη περίπτωση: $O(n)$ όταν όλα τα στοιχεία είναι ίσα

Πολυπλοκότητα χώρου: $O(\log n)$ λόγω της στοίβας αναδρομής