Πρώτη προγραμματιστική εργασία - Μάθημα Αλγορίθμων

Νικόλαος Σμυρνιούδης (3170148)

8 Μαρτίου 2019

1 Πρώτη άσκηση

Για να βρεθούν οι πρώτες και οι τελευτάιες θέσεις ενός στοιχείου x στον πίνακα πρώτα εκτελείται μια δυαδική αναζήτηση για να βρεθεί η θέση ένος στοιχείου απο το cluster που περιέχει τα x. Έπειτα με μια παραλλαγμένη μορφή δυαδικής αναζήτησης βρίσκονται και τα δύο άκρα του cluster. Ο χρόνος των αναζητήσεων περιγράφεται απο την παρακάτω εξίσωση γιατί σε ολους το αρχικό πρόβλημα σε ενα πινακα `A` ανάγεται σε $\mathcal{O}(1)$ χρόνο σε πρόβλημα για πίνακα `B` με τα μισά στοιχεία.

Συνολικά εκτελούνται 3 αναζητήσεις και για όλες ο χρόνος μπορεί να περιγραφεί με την αναδρομική εξίσωση:

$$T_n = T_{n/2} + \mathcal{O}(1) = \mathcal{O}(logn)$$

Εφόσον οι τρείς αναζητήσεις εκτελούνται σειριακά τότε ο συνολικός χρόνος θα είναι:

$$= \mathcal{O}(logn) + \mathcal{O}(logn) + \mathcal{O}(logn) = \mathcal{O}(logn)$$

Και συνεπώς ο συνολικός χρόνος είναι πολυπλοκότητας $\mathcal{O}(logn)$.

2 Δεύτερη άσκηση

Ο αλγόριθμος quickSort κανει partition στον αρχικό πίνακα με pivot ενα τυχαίο σημείο του πίνακα που υπολογίζεται ως $A((\text{high} - \text{low}) * x + \text{low})$ με x τυχαία μεταβλητη για την οποία ισχύει $x \in [0, 1]$.Η διαμέριση πρώτα αντιγράφει τα στοιχεια του A απο το low μεχρι το high σε έναν βοηθητικό πίνακα και ύστερα παραθέτει τα στοιχεία μικρότερα του pivot στην αριστερή πλευρά του αρχικού πίνακα, τα στοιχεία μεγαλύτερα του pivot στην δεξιά πλευρά του πίνακα και συμπληρώνει τις κενές θέσεις με τα στοιχεία ίσα με τον πίνακα. Αφου ολοκληρωθεί αυτή η διαδικασία ο αλγοριθμος quickSort συνεχίζει αναδρομικά στον πίνακα που περιέχει τα στοιχεία μικρότερα του pivot και αναδρομικά στον πίνακα που περιέχει τα στοιχεία μεγαλύτερα του pivot.

Ο αλγόριθμος QuickSort εφαρμοσμένος σε πίνακα με n στοιχεία θα εκτελέσει τον αλγόριθμο partition ο οποίος εκτελείται σε γραμμικό χρόνο και ύστερα τον αλγόριθμο quickSort σε δυό πίνακες m και k στοιχείων. Για αυτά τα m και k θα ισχύει πως $m + k < n$ αφού η partition χωρίζει τον πίνακα σε τρεις υποπίνακες, έναν με στοιχεία μικρότερα του pivot, έναν πίνακα με στοιχεία ίσα του pivot, και έναν πίνακα με στοιχεία μεγαλύτερα του pivot

Συνολικά ο χρόνος του αλγορίθμου μπορεί να περιγραφεί απο την αναδρομική εξίσωση:

$$T(n) = T(m) + T(k) + \mathcal{O}(n)$$

Στήν καλύτερη περίπτωση που πίνακας είναι σχεδόν γεμάτος με στοιχεία του pivot και τα m,k είναι $\mathcal{O}(1)$ η αναδρομική εξίσωση λύνεται ως

$$T(n) = T(m) + T(k) + \mathcal{O}(n) = \mathcal{O}(1) + \mathcal{O}(1) + \mathcal{O}(n) = \mathcal{O}(n)$$

Στην χειρότερη όμως περίπτωση το pivot επιλέγεται έτσι ώστε να είναι το μέγιστο στοιχείο ή το ελάχιστο στοιχείο του πίνακα και η αναδρομική εξίσωση θα γράφεται ώς

$$T(n) = T(m) + T(k) + \mathcal{O}(n) \leq T(n-1) + T(1) + \mathcal{O}(n) = \mathcal{O}(n^2)$$

3 Λεπτομέρειες Υλοποίησης

Η λύση για την πρώτη άσκηση περιέχεται στο αρχείο Exercise1.java ενώ για την δεύτερη στο αρχείο Exercise2.java. Τα προγράμματα λαμβάνουν τα δεδομένα εισόδου ως ένα αρχείο το οποίο δίνεται μέσω της γραμμής εντολών. Για παράδειγμα άν τα δεδομένα εισόδου βρίσκονται στο αρχείο data.txt τότε τα προγράμματα θα τρέξουν ως εξής

java part1 data.txt

java QuickSort data.txt

Εναλλακτικά υπάρχει και το αρχείο ExerciseSet1.java το οποίο τρέχει τις ασκήσεις με δεδομένα απο τα αρχεία 1.1-sm.txt και 1.2-sm.txt.