

Πρώτη εργασία στο μάθημα “Αλγόριθμοι και Προχωρημένες Δομές Δεδομένων”

Μεταπτυχιακό Πληροφορικής και Δικτύων

Γκόγκος Χρήστος

Τμήμα Πληροφορικής και Τηλεπικοινωνιών, Πανεπιστημίου Ιωαννίνων

Άρτα, Νοέμβριος 2021

1 Εισαγωγή

Η δυαδική αναζήτηση είναι ένας έξυπνος τρόπος εντοπισμού στοιχείων σε μια ακολουθία ταξινομημένων τιμών. Η ταχύτητα εκτέλεσης του αλγορίθμου είναι υψηλότερη σε σχέση με αλγορίθμους όπως η σειριακή αναζήτηση και η αναζήτηση με αναπηδήσεις. Στην παρούσα εργασία ζητείται, α) η συγγραφή αλγορίθμων σειριακής αναζήτησης, αναζήτησης με αναπηδήσεις και δυαδικής αναζήτησης, β) η εμπειρική μελέτη των αλγορίθμων, γ) ο εντοπισμός λαθών που μπορούν να υπάρξουν στην περιγραφή του αλγορίθμου δυαδικής αναζήτησης και δ) η διερεύνηση του κατά πόσο μπορεί να υπάρξει ταχύτερος αλγόριθμος αναζήτησης από τη δυαδική.

2 Περιγραφή προβλήματος

Το πρόβλημα που εξετάζεται στην παρούσα εργασία είναι η αναζήτηση μιας τιμής (key) σε μια ταξινομημένη σε αύξουσα σειρά ακολουθία τιμών (a) και η επιστροφή της θέσης στην οποία βρίσκεται. Οι αλγόριθμοι που θα χρησιμοποιηθούν είναι η σειριακή αναζήτηση, η αναζήτηση με αναπηδήσεις και η δυαδική αναζήτηση.

Ο αλγόριθμος σειριακής αναζήτησης εξετάζει τα στοιχεία της ακολουθίας στη σειρά (από αριστερά προς τα δεξιά) και σταματά είτε όταν εντοπίσει την τιμή key, είτε όταν έχει πλέον εξετάσει όλα τα στοιχεία της ακολουθίας. Ο αλγόριθμος αναζήτησης με αναπηδήσεις, εξετάζει τα στοιχεία της ακολουθίας με κάποιο βήμα step, δηλαδή εξετάζονται τα στοιχεία $a[0]$, $a[\text{step}]$, $a[2 \cdot \text{step}]$ κ.ο.κ. Αν εντοπίσει την τιμή key σταματά και επιστρέφει τη θέση στην οποία την βρήκε. Αν εντοπίσει τιμή που είναι μικρότερη από την τιμή key τότε επιστρέφει στην προηγούμενη θέση και εξετάζει σειριακά όλες τα στοιχεία της ακολουθίας από το $a[\text{step}+1]$ μέχρι και το $a[2 \cdot \text{step}-1]$ για την τιμή key. Ο αλγόριθμος δυαδικής αναζήτησης αποτελεί κλασσικό παράδειγμα αλγορίθμου που χρησιμοποιεί την τεχνική “Διαιρεί και Βασίλευε”. Διαιρεί επαναληπτικά την ακολουθία σε δύο υποακολουθίες με βάση το στοιχείο που εντοπίζεται στη μέση της ακολουθίας, εξετάζει το στοιχείο αυτό και εάν η τιμή key δεν εντοπιστεί στη θέση αυτή συνεχίζει είτε με το αριστερό τμήμα της ακολουθίας είτε με το δεξί τμήμα της ακολουθίας. Η διαδικασία συνεχίζεται μέχρι είτε να εντοπιστεί το στοιχείο key είτε η ακολουθία που εξετάζεται να μην έχει πλέον στοιχεία, οπότε και συμπεραίνεται ότι η τιμή key δεν υπάρχει στην ακολουθία τιμών.

2.1 Υλοποίηση αλγορίθμων

Ζητείται η υλοποίηση των αλγορίθμων σειριακής αναζήτησης (serial_search), αναζήτησης με αναπηδήσεις (jump_search) και δυαδικής αναζήτησης (binary_search). Κάθε αλγόριθμος θα πρέπει να δέχεται ως παράμετρο μια ακολουθία ταξινομημένων τιμών (a) και μια τιμή προς αναζήτηση (key). Ειδικά για την περίπτωση του αλγορίθμου jump_search θα υπάρχει και μια παράμετρος step με προκαθορισμένη τιμή 100. Και οι τρεις αλγόριθμοι θα πρέπει να επιστρέφουν είτε τη θέση εμφάνισης του στοιχείου key στην ακολουθία τιμών, είτε την τιμή -1 αν το key δεν υπάρχει στην ακολουθία τιμών. Αν η τιμή key υπάρχει πολλές φορές αρκεί να επιστραφεί μια από τις θέσεις εμφάνισής της.

2.2 Εμπειρική μελέτη αλγορίθμων αναζήτησης

Χρονομετρήστε τους αλγορίθμους αναζήτησης για διάφορα μεγέθη ακολουθιών ακεραίων τιμών που θα επιλέξετε. Κατασκευάστε ένα γράφημα που να απεικονίζει τα αποτελέσματα (πιθανά να χρειαστεί να χρησιμοποιήσετε λογαριθμικούς άξονες στο γράφημα λόγω της μεγάλης απόκλισης στους χρόνους εκτέλεσης των αλγορίθμων).

2.3 Προβλήματα στην περιγραφή του αλγορίθμου δυαδικής αναζήτησης

Στο άρθρο [Pat88] περιγράφονται λάθη που συναντώνται σύμφωνα με τον συγγραφέα του άρθρου σε βιβλία και έχουν να κάνουν με τη δυαδική αναζήτηση. Με συντομία περιγράψτε δύο τουλάχιστον από τα λάθη που αναφέρει το άρθρο.

2.4 Υπάρχει ταχύτερος αλγόριθμος αναζήτησης από τη δυαδική αναζήτηση;

Σχολιάστε σχετικά με το εάν μπορεί να υπάρξει αλγόριθμος αναζήτησης που να είναι ταχύτερος από τη δυαδική αναζήτηση.

3 Παραδοτέα εργασίας

Τα παραδοτέα της εργασίας θα είναι τα ακόλουθα:

1. Κώδικας που υλοποιεί τους αλγορίθμους που ζητήθηκαν.
2. Unit Tests ελέγχου της ορθότητας των αλγορίθμων.
3. Πρόγραμμα οδηγός για τα πειράματα χρονομέτρησης.
4. Οδηγίες εκτέλεσης του κώδικα.
5. Τεχνική αναφορά για την εργασία, στα πρότυπα σύντομου επιστημονικού άρθρου. Η αναφορά θα πρέπει να είναι περίπου 2 σελίδες και να περιέχει:
 - (α') Τα χαρακτηριστικά του υπολογιστή που χρησιμοποιήθηκε στα πειράματα (CPU, RAM, ...).
 - (β') Το λογισμικό που χρησιμοποιήθηκε (π.χ. Python 3.10, GCC, VsCode, Excel, gnuplot, matplotlib, ...).
 - (γ') Τις μετρήσεις χρόνου εκτέλεσης του αλγορίθμου και τη θεωρητική πολυπλοκότητα των αλγορίθμων.
 - (δ') Γραφική σύγκριση μέσω διαγραμμάτων των αποτελεσμάτων που παράγουν οι αλγόριθμοι.
 - (ε') Το σχολιασμό σας σχετικά με τα θέματα των παραγράφων 2.3 και 2.4.

4 Παρατηρήσεις

- Η υλοποίηση του κώδικα θα πρέπει να γίνει κατά προτίμηση στη γλώσσα προγραμματισμού Python. Εναλλακτικά μπορεί να χρησιμοποιηθεί C, C++, C#, Java, JavaScript.
- Η εργασία είναι ατομική και η παράδοσή της γίνεται στο ecourse του μαθήματος μέχρι τις 19/12/2021.

Αναφορές

[Pat88] Richard E Pattis. Textbook errors in binary searching. *ACM SIGCSE Bulletin*, 20(1):190–194, 1988.