



ΠΑΝΕΠΙΣΤΗΜΙΟ
ΠΑΤΡΩΝ
UNIVERSITY OF PATRAS

Αλγόριθμοι και Δομές Δεδομένων

PROJECT 4

ΧΡΗΣΤΑΚΗΣ ΜΑΚΑΡΙΟΣ | ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΑΤΡΩΝ | ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ
ΜΗΧΑΝΙΚΩΝ & ΤΕΧΝΟΛΟΓΙΑΣ ΥΠΟΛΟΓΙΣΤΩΝ

1 Διαδικασία

Ο συνημμένος κώδικας αναπτύχθηκε σε γλώσσα Python(3.6.5) και υλοποιεί 4 διαφορετικούς αλγόριθμους για την αναζήτηση κοινών τιμών μεταξύ δύο τυχαία δημιουργημένων σετ τιμών τα οποία έχουν κοινά τα μισά τους στοιχεία και τα υπόλοιπα τυχαία.

Αλγόριθμοι που υλοποιήθηκαν:

- Γραμμική αναζήτηση (Linear Search)
- Αναζήτηση με χρήση hash table που χρησιμοποιεί open addressing ως μέθοδο αντιμετώπισης συγκρούσεων.
- Αναζήτηση με χρήση hash table που χρησιμοποιεί chaining ως μέθοδο αντιμετώπισης συγκρούσεων.
- Ταξινόμηση του ενός πίνακα χρησιμοποιώντας τον αλγόριθμο merge sort και έπειτα δυαδική αναζήτηση (binary search) στον ταξινομημένο πίνακα για τα στοιχεία του άλλου.

2 Σύντομος σχολιασμός του κώδικα

2.1 Συνάρτηση makesets

Η συνάρτηση αυτή παίρνει ως όρισμα έναν ακέραιο αριθμό N και δημιουργεί 2 σετ (λίστες) αριθμών με τα εξής χαρακτηριστικά:

- a) Τα μισά στοιχεία είναι ακριβώς τα ίδια.
- b) Κάθε στοιχείο είναι ακέραιος αριθμός από το 0 μέχρι το $10 \cdot N$.
- c) Οι ακέραιοι αριθμοί αυτοί είναι τυχαία δημιουργημένοι χρησιμοποιώντας τη βιβλιοθήκη random της Python και θέτοντας ως seed τον αριθμό μητρώου μου (εμφανίζεται στο εξώφυλλο).

Τέλος επιστρέφει τα δημιουργηθέντα σετ αριθμών για περαιτέρω χρήση.

2.2 Συνάρτηση linsearch

Η συνάρτηση αυτή παίρνει ως όρισμα δύο λίστες αριθμών και επιστρέφει τον αριθμό των κοινών στοιχείων του πρώτου σετ αριθμών στο δεύτερο σετ, δηλαδή πόσοι αριθμοί του σετ 1 υπάρχουν και στο σετ 2.

Η καταμέτρηση των κοινών αριθμών γίνεται με τον απλό αλγόριθμο της γραμμικής αναζήτησης.

2.3 Συνάρτηση hash_searches

Στη συνάρτηση αυτή αρχικά κατασκευάζονται οι 2 hash table, ένας χρησιμοποιώντας open addressing και ένας chaining, μέσω των αντίστοιχων συναρτήσεων. Αφού θέλουμε να ψάξουμε ποιά στοιχεία του 1^{ου} σετ βρίσκονται στο 2^ο τοποθετούμε στους hash tables τα στοιχεία του δεύτερου σετ δεδομένων.

Έπειτα διατρέχοντας τον πίνακα με τα στοιχεία του πρώτου σετ αριθμών ελέγχουμε ένα ένα αν υπάρχουν μέσα στους δημιουργηθέντες hash table χρησιμοποιώντας τις αρμόζουσες συναρτήσεις για κάθε περίπτωση (open addressing ή chaining) και αυξάνουμε τους αντίστοιχους counter που μετρούν πόσα κοινά στοιχεία υπάρχουν για κάθε μέθοδο αντιμετώπισης των collision.

Τέλος επιστρέφονται οι μετρηθέντες τιμές για κάθε μέθοδο.

Σημειώσεις:

- i. Στον χρόνο που εμφανίζεται για κάθε μεθοδο (chaining, open addressing) περιλαμβάνεται και ο χρόνος δημιουργίας των hash table.
- ii. Για το hashing των τιμών του πίνακα 2 (ακέραιοι αριθμοί) χρησιμοποιήθηκε η hash function της προηγούμενης εργασίας (εργασία 3), η οποία χρησιμοποιεί το κάθε ψηφίο του ακεραίου αυτού ως συντελεστή ενός πολυωνύμου του x και επιστρέφει την τιμή αυτού στο $x = 33$ μέσω του κανόνα του Horner.

2.4 Συνάρτηση bin_count

Η συνάρτηση αυτή αρχικά χρησιμοποιεί τη συνάρτηση merge_sort, η οποία υλοποιεί τον γνωστό αλγόριθμο ταξινόμησης αναδρομικά, για να ταξινομήσει το 2^ο σετ αριθμών.

Έπειτα, διατρέχοντας τα στοιχεία του 1^{ου} σετ αριθμών, χρησιμοποιεί την συνάρτηση binsearch, η οποία υλοποιεί τον αλγόριθμο δυαδικής αναζήτησης, για να προσδιορίσει εάν το εκάστοτε στοιχείο του 1^{ου} σετ αριθμών βρίσκεται και στο 2^ο και αυξάνει τον αντίστοιχο counter για την μέθοδο αυτή.

Στον χρόνο που εμφανίζει περιλαμβάνεται και ο χρόνος ταξινόμησης του σετ δεδομένων.

Τέλος επιστρέφει τον παραπάνω μετρητή για εμφάνιση του.

3 Αποτελέσματα

Το αποτέλεσμα της εκτέλεσης για $N = 10.000$ φαίνεται παρακάτω:

```
Number Set size (N) is: 10000
Linear Search ran for 4.8652475 sec.
Linear Search found 5472 common numbers
-----
Open Addressing Collisions: 2603
Open addressing Search ran for 0.0634003 sec.
Chaining Collisions : 2261
Chaining Search ran for 0.0624001 sec.
Hashing with Open addressing search found: 5472 common numbers
Hashing with Chaining search found 5472 common numbers
-----
Binary Search ran for 0.1248002 sec.
Merge Sort and then Binary Search found 5472 common numbers
Process returned 0 (0x0)          execution time : 5.191 s
Press any key to continue . . . _
```

Όπως ήταν αναμενόμενο όλοι οι αλγόριθμοι βρήκαν τον ίδιο αριθμό κοινών στοιχείων μεταξύ των δυο σετ αριθμών. Οι αλγόριθμοι που χρησιμοποιούν hash table είναι οι πιο γρήγοροι, αφού στη μέση περίπτωση είναι $O(1)$, έπειτα ακολουθεί ο αλγόριθμος με merge sort + binary search, ο οποίος είναι $O(n * \log n)$ [συμπεριλαμβάνοντας το sorting] και τέλος ο πιο αργός είναι η γραμμική αναζήτηση που είναι $O(N^2)$.