

1η εργασία στο μάθημα
“Αλγόριθμοι και Πολυπλοκότητα”
Τμήμα Πληροφορικής και Τηλεπικοινωνιών (Άρτα)
Πανεπιστήμιο Ιωαννίνων

Χρήστος Γκόγκος
<http://chgogos.github.io>

17 Οκτωβρίου 2023



Γρήγορος υπολογισμός συγκεντρωτικών χρεώσεων πιστωτικών καρτών

Η παρούσα εργασία αφορά την υλοποίηση μιας γρήγορης λύσης σε ένα πρόβλημα υπολογισμού συνολικών χρεώσεων για κατόχους πιστωτικών καρτών.

Ερώτημα 1 (2 μονάδες)

Δημιουργήστε στη μνήμη του υπολογιστή 1.000.000 υποθετικές χρεώσεις πιστωτικών καρτών με την ακόλουθη διαδικασία.

1. Αρχικά δημιουργήστε 20.000 διαφορετικούς μεταξύ τους τυχαίους αριθμούς πιστωτικών καρτών με κάθε αριθμό πιστωτικής κάρτας να αποτελείται από 16 ψηφία (π.χ. 1234-5678-9012-3456, χωρίς τις παύλες).
2. Για 1.000.000 επαναλήψεις, επιλέγοντας με τυχαίο τρόπο αριθμό πιστωτικής κάρτας από τους διαθέσιμους να δημιουργείτε μια χρέωση τη φορά για ένα ποσό τυχαία επιλεγμένο στο διάστημα από 10 μέχρι 1.000 ευρώ (με 2 δεκαδικά ψηφία).

Παρατήρηση Αρχικοποιήστε τη διαδικασία παραγωγής τυχαίων τιμών με σπόρο (seed) τον αριθμό μητρώου σας.

Ερώτημα 2 (3 μονάδες)

Επεξεργαστείτε τα δεδομένα χρεώσεων που δημιουργήσατε στο προηγούμενο ερώτημα και υπολογίστε με τον ταχύτερο δυνατό τρόπο, χρησιμοποιώντας οποιαδήποτε διαθέσιμη βιβλιοθήκη της γλώσσας προγραμματισμού που χρησιμοποιείτε, τα ακόλουθα:

1. Την κάρτα με το μικρότερο συνολικό ποσό πληρωμών.
2. Την κάρτα με το μεγαλύτερο συνολικό ποσό πληρωμών.
3. Την κάρτα με το μικρότερο πλήθος συναλλαγών.
4. Την κάρτα με το μεγαλύτερο πλήθος συναλλαγών.

Χρονομετρήστε την εκτέλεση όλων των λειτουργιών συνολικά.

Ερώτημα 3 (3 μονάδες)

Υλοποιήστε έναν δικό σας πίνακα κατακερματισμού με ανοικτή διευθυνσιοδότηση και γραμμική διερεύνηση (open addressing με linear probing). Ο πίνακας να ξεκινά με μέγεθος ίσο με τον πρώτο αριθμό 101. Αν ο συντελεστής φόρτωσης (load factor) του πίνακα κατακερματισμού ξεπερνά το 70% τότε το μέγεθός του να γίνεται ίσο με τον μικρότερο πρώτο αριθμό που είναι διπλάσιος του προηγούμενου μεγέθους (π.χ. μετά τον πρώτο διπλασιασμό το μέγεθος του πίνακα κατακερματισμού θα είναι 211, στον επόμενο διπλασιασμό θα είναι 431 κ.ο.κ.).

Παρατήρηση Επιλέξτε κατάλληλη συνάρτηση κατακερματισμού, συμβουλευτείτε το [1].

Ερώτημα 4 (2 μονάδες)

Επαναλάβετε το ερώτημα 2 χρησιμοποιώντας τον πίνακα κατακερματισμού που υλοποιήσατε στο ερώτημα 3. Ποιος είναι ο αριθμός των συγκρούσεων που προκαλούνται κατά την εισαγωγή των συναλλαγών στο πίνακα κατακερματισμού; Θεωρείστε ως σύγκρουση την απόπειρα εισαγωγής ενός κλειδιού στον πίνακα κατακερματισμού που θα τοποθετούσε αρχικά το κλειδί σε θέση ήδη κατειλημμένη από άλλο κλειδί.

Χρονομετρήστε την εκτέλεση όλων των λειτουργιών όπως και στο ερώτημα 2. Συγκρίνατε τους χρόνους και βελτιώστε την υλοποίησή σας μέχρι οι χρόνοι εκτέλεσης να βρίσκονται “κοντά”. Τρέξτε ξανά τον αλγόριθμο για 2.000.000, 4.000.000 και 6.000.000 χρεώσεις. Κατασκευάστε ένα γράφημα με τους χρόνους που απαιτεί η δική σας υλοποίηση του πίνακα κατακερματισμού και η υλοποίηση του ερωτήματος 2. Εκτιμήστε την πολυπλοκότητα των αλγορίθμων που υλοποιήσατε με βάση το γράφημα. Συντάξτε μια μικρή αναφορά με αποτελέσματα και συμπεράσματα ή παρατηρήσεις για την εργασία συνολικά.

Υποβολή εργασίας

Σχετικά με την υποβολή της εργασίας ισχύουν τα ακόλουθα:

- Η εργασία μπορεί να υποβληθεί μόνο στο ecourse: <https://ecourse.uoi.gr/user/index.php?id=1946>.
- Ως προθεσμία υποβολής της εργασίας έχει οριστεί η 30/11/2023.
- Η εργασία είναι ατομική.
- Η εργασία μπορεί να υλοποιηθεί σε Python, C, C++ ή Java.
- Ο κώδικας της εργασίας πρέπει να ανέβει στο GitHub σε ιδιωτικό αποθετήριο.
- Παραδοτέα εργασίας (σε ένα zip αρχείο):
 - Αναφορά με απαντήσεις στα ερωτήματα της εργασίας.
 - Κώδικας με επαρκή σχολιασμό και οδηγίες εκτέλεσης.

Αναφορές

- [1] Arash Partow. *General Purpose Hash Function Algorithms*. <https://www.partow.net/programming/hashfunctions/index.html>.