

# ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΑΤΡΩΝ

ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΤΕΧΝΟΛΟΓΙΑΣ ΥΠΟΛΟΓΙΣΤΩΝ

## ΑΛΓΟΡΙΘΜΟΙ ΚΑΙ ΔΟΜΕΣ ΔΕΔΟΜΕΝΩΝ

Άσκηση: Υλοποίηση Hash Table

Πάυλος Λουκαρέας

A.M. 1046970

### Εισαγωγή

Στην εργασία αυτή γίνεται η εύρεση η εύρεση των κωδικών των πελατών μιας εταιρείας πιστωτικών καρτών που ξοδεύουν περισσότερο και ψωνίζουν συχνότερα από τους υπόλοιπους πελάτες, με χρήση πίνακα κατακερματισμού. Οι αγορές που γίνονται μέσω της τράπεζας είναι ένα εκατομμύριο, οι κωδικοί των καρτών είναι δεκαεξαψήφιοι (τυχαίοι με προκαθορισμένο τρόπο) και το ποσό κάθε αγοράς κυμαίνεται τυχαία από 10 έως 1000 ευρώ. Η χρήση πίνακα κατακερματισμού γίνεται για την ενημέρωση της βάσης δεδομένων των πελατών ώστε να εξαχθούν στο τέλος τα στοιχεία που αφορούν αυτούς. Το πλεονέκτημα του πίνακα κατακερματισμού οφείλεται στην ταχεία εύρεση των στοιχείων των πελατών μέσω της συνάρτησης κατακερματισμού, ενώ οι συγκρούσεις που προκύπτουν αντιμετωπίζονται μέσω της τεχνικής Open Addressing. Το πρόγραμμα υλοποιήθηκε σε γλώσσα Python και δομήθηκε σε τρεις συναρτήσεις οι οποίες καλούνται μέσα από τη συνάρτηση `main()`.

Η συνάρτηση `main()` φαίνεται παρακάτω έχει το ρόλο της κλήσης των υπολοίπων συναρτήσεων, της διαβίβασης των αποτελεσμάτων που αυτές παράγουν και της παρουσίασης του αποτελέσματος που ζητείται.

```
def main():
    for i in range(100000):
        customer, customer_ID, price = createSale()
        store_hash(customer, customer_ID, price)
    max_spender_shopper(h_table)
```

Για την ευκολότερη διαχείριση των δεδομένων υλοποιήθηκε μια κλάση για τις πιστωτικές κάρτες, τις οποίες τα αντικείμενα αποθηκεύονται στον πίνακα κατακερματισμού. Σε κάθε αντικείμενο που δημιουργείται αποθηκεύονται ο κωδικός της κάρτας καθώς και το ποσό με το οποίο έγινε η πρώτη αγορά. Επιπλέον δημιουργήθηκε και μια μέθοδος για την ενημέρωση των δεδομένων που απαιτείται κάθε φορά που γίνεται μια επόμενη αγορά από τον ίδιο πελάτη.

Μέσα στο βρόχο επανάληψης δημιουργείται κάθε φορά μια αναπαράσταση σε ακέραιο αριθμό του κωδικού της πιστωτικής κάρτας, ο κωδικός της πιστωτικής κάρτας σε αλφαριθμητικό και το ποσό της συγκεκριμένης αγοράς. Έπειτα τα δεδομένα αυτά, με βάση τον δείκτη που επιστρέφει η συνάρτηση κατακερματισμού, αποθηκεύονται στον hash table έως ότου τελειώσουν όλες οι επαναλήψεις. Λόγω της ύπαρξης 43680 διαφορετικών καρτών χρειαζόμαστε έναν πρώτο αριθμό κοντά στο  $2 \cdot 4380 = 87360$  ο οποίος είναι ο 87383 για το μέγεθος του πίνακα κατακερματισμού καθώς και για τη συνάρτηση κατακερματισμού.

Για τον αλφαριθμητικό κωδικό της κάρτας διαλέγονται τυχαία τέσσερις διαφορετικές θέσεις όπου αντικαθίστανται με τους χαρακτήρες A, B, C, D. Η μετατροπή σε ακέραιο γίνεται μέσω ενός αθροίσματος του γινομένου της τιμής στον πίνακα ASCII για κάθε χαρακτήρα, της θέσης του ως δύναμη στη βάση ενός πρώτου αριθμού.

Η αποθήκευση στον πίνακα γίνεται μέσω της τεχνικής Open Addressing. Αρχικά υπολογίζεται η παράσταση  $\text{int\_ID} \bmod N_{\text{table}}$  και το αποτέλεσμα είναι η θέση του πίνακα που θα προσπελάσουμε. Αν είναι κενή, τότε φτιάχνουμε ένα αντικείμενο κάρτας. Αν δεν είναι κενή τότε αναζητούμε με γραμμικό τρόπο τα επόμενα στοιχεία για να φτιάξουμε νέα αντικείμενα ή να αναβαθμίσουμε τα στοιχεία των υπαρχόντων.

Τέλος, αφού τελειώσουν όλες οι επαναλήψεις γίνεται μια γραμμική αναζήτηση για τους καταναλωτές που ξόδεψαν τα περισσότερα και ψώνισαν συχνότερα με τις κάρτες τους.

Τα αποτελέσματα για την προσομοίωση που έγινε είναι:

```
The customer D2345ACB90123456 owes the most, with a dept of 24817
```

```
The customer B2C45D789012345A shops the most, with usages of 45
```