

Αλγόριθμοι και προχωρημένες δομές δεδομένων
ΣΕΤ ΑΣΚΗΣΕΩΝ #3

Ημερομηνία παράδοσης 10 Ιανουαρίου 2019

Χρυσόστομος Στύλιος και Χρήστος Γκόγκος

14/12/2018

Άσκηση 1

Στο μάθημα του <https://stepik.org/> με τίτλο “Data Structures (by Niema Moshiri και Liz Izhikevich)”, ολοκληρώστε συμπεριλαμβανομένων των κουίζ και των ασκήσεων τις ακόλουθες ενότητες:

- 5. Hashing
- 6. Implementing a Lexicon
 - 6.1 Creating a Lexicon
 - 6.2 Using Linked Lists
 - 6.3 Using Arrays
 - 6.5 Using Hash Tables and Hash Maps

Συγκεντρώστε τουλάχιστον 30 πόντους από τις δύο ενότητες.

Άσκηση 2

Τα Bloom Filters [1] είναι δομές δεδομένων που υλοποιούν σύνολα (sets). Αν τεθεί σε ένα Bloom Filter η ερώτηση: “Υπάρχει το στοιχείο x μέσα στο σύνολο A ;” οι πιθανές απαντήσεις που μπορούν να ληφθούν είναι *όχι* ή *ίσως*. Αν η απάντηση είναι *όχι* τότε είμαστε σίγουροι ότι το στοιχείο x δεν υπάρχει στο σύνολο A . Αν η απάντηση είναι *ίσως* τότε το στοιχείο x πιθανά υπάρχει στο σύνολο A .

Μια ενδιαφέρουσα οπτικοποίηση των Bloom Filters μπορεί να βρεθεί στο <https://www.jasondavies.com/bloomfilter/>.

Εκφώνηση άσκησης Έστω ότι επιθυμούμε να ελέγξουμε για 10.000.000 ακέραιες τιμές πόσες από αυτές δεν περιέχονται σε ένα δεδομένο σύνολο 10.000 ακεραίων τιμών (σύνολο A) που ανήκουν στο διάστημα $[0, 100.000]$. Επιλύστε το πρόβλημα χρησιμοποιώντας 4 προσεγγίσεις, χρονομετρήστε την κάθε μια και καταγράψτε τα αποτελέσματα.

1. Στην πρώτη προσέγγιση χρησιμοποιήστε ένα `std::vector`, αποθηκεύστε σε αυτό τις τιμές του συνόλου A , ταξινομήστε το και αναζητήστε τις τιμές χρησιμοποιώντας δυαδική αναζήτηση.
2. Στη δεύτερη προσέγγιση χρησιμοποιήστε ένα `std::set` (υλοποιείται στην STL ως δυαδικό δένδρο αναζήτησης) για τις τιμές του συνόλου A .
3. Στην τρίτη προσέγγιση χρησιμοποιήστε ένα `std::unordered_set` (υλοποιείται στην STL ως πίνακας κατακερματισμού) για τις τιμές του συνόλου A .
4. Στην τέταρτη προσέγγιση υλοποιήστε ένα Bloom Filter για τις τιμές του συνόλου A .

Αναφορές

- [1] Bloom, Burton H. Space/time trade-offs in hash coding with allowable errors. Communications of the ACM, 13;7, pp.422-426, 1970.