

HY240: Δομές Δεδομένων

Εαρινό Εξάμηνο – Ακαδημαϊκό Έτος 2019-20

Παναγιώτα Φατούρου

2^ο Σετ Ασκήσεων

Ημερομηνία Παράδοσης: Δευτέρα, 9 Μαρτίου 2020

Τρόπος Παράδοσης: Οι ασκήσεις μπορούν να παραδοθούν είτε σε ηλεκτρονική μορφή χρησιμοποιώντας το πρόγραμμα turnin (δείτε οδηγίες στην ιστοσελίδα του μαθήματος), είτε στους βοηθούς του μαθήματος στο εργαστήριο των μεταπτυχιακών, τη Δευτέρα, 9 Μαρτίου 2020, ώρα 13:00-14:00. Ασκήσεις που παραδίδονται μετά τις 14:00 της Δευτέρας, 9/3/2020 θεωρούνται εκπρόθεσμες. Εκπρόθεσμες ασκήσεις γίνονται δεκτές σε ηλεκτρονική μορφή (pdf) και η αποστολή τους πρέπει να γίνει χρησιμοποιώντας το πρόγραμμα turnin. Εναλλακτικά, εκπρόθεσμες ασκήσεις μπορούν να παραδοθούν στη διδάσκουσα.

Άσκηση 1 [40 Μονάδες]

Έστω ότι μια βιβλιοθήκη παρέχει υλοποιήσεις στοιβών και ουρών ακεραίων. Η βιβλιοθήκη επιτρέπει να ορίσετε μια στοιβή ή/και μια ουρά και να καλέσετε τις 5 βασικές λειτουργίες στη στοιβή ή στην ουρά που ορίσατε. Για παράδειγμα, ο ορισμός μιας στοιβας S1 γίνεται με τη δήλωση: Stack S1, ενώ μιας ουράς Q1 γίνεται με τη δήλωση: Queue Q1.

Οι λειτουργίες που υποστηρίζονται στη στοιβή είναι οι εξής: (1) void MakeEmptyStack(Stack S), (2) boolean IsEmptyStack(Stack S), (3) int Top(Stack S), (4) int Pop(Stack S), και (5) void Push(Stack S, int x). Αντίστοιχα, οι λειτουργίες που υποστηρίζονται στην ουρά είναι οι εξής: (1) void MakeEmptyQueue(Queue Q), (2) boolean IsEmptyQueue(Queue Q), (3) int Front(Queue Q), (4) int Dequeue(Queue Q), και (5) void Enqueue(Queue Q, int x)

1. Ταξινόμηση ακολουθίας ακέραιων αριθμών [20M]

Σχεδιάστε αλγόριθμο ο οποίος θα διαβάσει μια ακολουθία από ακεραίους από το πληκτρολόγιο (η ακολουθία αυτή δεν είναι απαραίτητα ταξινομημένη) και θα χρησιμοποιεί ουρές ή στοιβές για να τυπώσει τους χαρακτήρες αυτούς σε φθίνουσα διάταξη. Ο αλγόριθμος θα πρέπει να χρησιμοποιεί το μικρότερο δυνατό πλήθος ουρών ή/και στοιβών.

2. Προθεματική μορφή αριθμητικών εκφράσεων και αποτίμηση αυτών [20M]

Οι αριθμητικές εκφράσεις αποτελούνται από τελεστές και μεταβλητές ή τελεστέους. Στην άσκηση αυτή εστιάζουμε σε τελεστές που είναι δυαδικοί (δηλαδή εφαρμόζονται σε δύο μεταβλητές). Ενδοθεματική (infix) ονομάζεται η μορφή παράστασης αριθμητικών εκφράσεων στην οποία ο τελεστής εμφανίζεται μεταξύ των δύο μεταβλητών. Στην μεταθεματική (postfix) μορφή παράστασης κάθε τελεστής εμφανίζεται μετά το ζεύγος των αντίστοιχων μεταβλητών, ενώ στην προθεματική ο τελεστής εμφανίζεται πριν το ζεύγος των δύο μεταβλητών.

Για παράδειγμα, η αριθμητική έκφραση $((a+(b*c))+((d+e)*f)-(g/h))+i$ είναι σε ενδοθεματική μορφή. Η προθεματική ισοδύναμη έκφραση αυτής είναι $+++a*bc-*+def/ghi$. Ζητείται αλγόριθμος που θα παίρνει ως είσοδο την προθεματική μορφή μιας αριθμητικής έκφρασης και θα αποτιμά την έκφραση (δηλαδή θα τυπώνει την τιμή της έκφρασης).

Οι αλγόριθμοι και για τα δύο ερωτήματα επιτρέπεται να χρησιμοποιούν μόνο στοιβές και ουρές ως βοηθητικές δομές.

Άσκηση 2 [20 Μονάδες]

Μια συμμορία από n ληστές στην Άγρια Δύση περικυκλώνεται από το απόσπασμα του σερίφη. Οι ληστές έχουν μόνο ένα άλογο για να διαφύγουν. Έτσι συμφωνούν στον εξής αλγόριθμο με σκοπό την επιλογή του τυχερού που θα πάρει το άλογο και θα σωθεί. Οι ληστές σχηματίζουν έναν κύκλο, ο αρχηγός αντιστοιχίζεται στον αριθμό 1 και η αρίθμηση συνεχίζεται ωρολογιακά. Σε ένα καπέλο τοποθετούν n λαχνούς με αριθμούς από το 1 μέχρι το n . Ο αρχηγός τραβάει ένα λαχνό με τον αριθμό, έστω i . Αυτό σημαίνει ότι ο i -οστός ληστής πρέπει να εξαιρεθεί. Η διαδικασία αυτή των εξαιρέσεων συνεχίζεται θεωρώντας για αρχή της καινούριας αρίθμησης τον επόμενο ληστή αυτού που μόλις εξαιρέθηκε. Αυτός που μένει τελευταίος παίρνει το άλογο, όλα τα κλοπιμαία και φεύγει.

Ζητείται να υλοποιηθεί ο αλγόριθμος αυτός χρησιμοποιώντας συνδεδεμένες λίστες. Εξηγήστε τι είδους λίστα θα χρησιμοποιήσετε για την υλοποίηση. Η είσοδος στον αλγόριθμο είναι ένας δείκτης στο στοιχείο μιας λίστας η οποία περιέχει ένα στοιχείο για κάθε ληστή και το στοιχείο στο οποίο δείχνει ο δείκτης αντιστοιχεί στον αρχηγό. Τα στοιχεία είναι τοποθετημένα με τη σειρά από το 1 ως το n . Κάθε στοιχείο της λίστας αυτής περιέχει επίσης το όνομα του αντίστοιχου ληστή. Η έξοδος του αλγορίθμου θα είναι τα ονόματά τους καθώς θα βγαίνουν από το παιχνίδι. Για κάθε ληστή που βγαίνει από το παιχνίδι, ο αντίστοιχος κόμβος της λίστας πρέπει να διαγράφεται.

Άσκηση 3 [40 Μονάδες]

Θεωρήστε τη λειτουργία $\text{Equal}(S_1, S_2)$ (ισότητα συνόλων), όπου S_1 και S_2 είναι δύο σύνολα. Η $\text{Equal}()$ πρέπει να παίρνει ως όρισμα ένα δείκτη $L1$ στο πρώτο στοιχείο της λίστας που υλοποιεί το σύνολο S_1 και ένα δείκτη $L2$ στο πρώτο στοιχείο της λίστας που υλοποιεί το S_2 . Θα πρέπει να επιστρέφει TRUE αν το S_1 είναι το ίδιο με το σύνολο S_2 και FALSE διαφορετικά. Οι λίστες που αναπαριστούν τα S_1 και S_2 θα πρέπει να παραμένουν αναλλοίωτες κατά την εκτέλεση του αλγορίθμου σας.

1. Ζητείται αλγόριθμος που να υλοποιεί τη λειτουργία $\text{Equal}()$ επί των συνόλων, δεδομένου ότι τα σύνολα υλοποιούνται με **ταξινομημένες απλά-συνδεδεμένες λίστες**. **Ο αλγόριθμος σας θα πρέπει να έχει πολυπλοκότητα $O(n_1 + n_2)$** , όπου n_1 και n_2 είναι το πλήθος στοιχείων των συνόλων L_1 και L_2 , αντίστοιχα. Εξηγήστε γιατί ο αλγόριθμός σας επιτυγχάνει την πολυπλοκότητα αυτή. **[20M]**
2. Ζητείται αλγόριθμος που να υλοποιεί τη λειτουργία $\text{Equal}()$ επί των συνόλων, δεδομένου ότι τα σύνολα υλοποιούνται με **μη-ταξινομημένες απλά-συνδεδεμένες λίστες**. Ποια είναι η πολυπλοκότητα του αλγορίθμου σας; **[20M]**