ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ
ATHENS UNIVERSITY OF ECONOMICS AND BUSINESS
Σταμάτης Χουλιαράς
3190221
Ιωάννης Βουσβουκής
3190027

Δομές Δεδομένων (Εργασία 1)

Θέμα Α:

Αρχικά δημιουργήσαμε μια ξεχωριστή κλάση Node , που βοηθά στην κατασκευή κόμβων για την υλοποίηση διπλά συνδεδεμένης λίστας. Έπειτα χρησιμοποιώντας τις αφηρημένες μεθόδους της StringDouble EndedQueue φτιάξαμε τις ζητούμενες λειτουργίες.

Αναλυτικότερα, οι μέθοδοι εισαγωγής/εξαγωγής δημιουργούν έναν νέο κόμβο με χρήση generics και μεταβάλλουν τους δείκτες header-tail χωρίς περαιτέρω επεξεργασία στην υπόλοιπη λίστα. Έτσι επιτυγχάνουν την χρονική πολυπλοκότητα 0(1).

Η size() λειτουργεί ως εξής:

Κάθε φορά που πραγματοποιείται μια λειτουργία εισαγωγής/εξαγωγής μεταβάλλεται ένας μετρητής. Με την κλήση της συνάρτησης αυτής απλά επιστρέφεται ο μετρητής επιτυγχάνοντας χρονική πολυπλοκότητα Ο(1).

Θέμα Β:

Αρχικά δημιουργήσαμε δύο μεθόδους που βοηθούν στον έλεγχο εγκυρότητας της εισόδου του χρήστη. Αυτές είναι οι εξής:

1. Η isNumeric που δέχεται έναν χαρακτήρα και επιστρέφει true ή false ανάλογα αν είναι ο χαρακτήρας αριθμός ή όχι.
2. Η isAcceptable που δέχεται μία ακολουθία string (στη συγκεκριμένη περίπτωση μία postfix παράσταση) και επιστρέφει true αν αυτή είναι αποδεκτή.

Ο αλγόριθμος λειτουργεί ως εξής:

Στην αρχή καλείται η μέθοδος isAcceptable ώστε να ελεγχθεί η εγκυρότητα της εισόδου του χρήστη. Στη συνέχεια δημιουργούμε μια λίστα και έπειτα χρησιμοποιούμε ένα for loop το οποίο σαρώνει κάθε στοιχείο της ακολουθίας και για κάθε ένα καλεί την μέθοδο IsNumeric. Αν η μέθοδος επιστρέψει true τότε το στοιχείο είναι operand και θα το εισάγουμε στην λίστα αλλιώς το στοιχείο είναι operator. Σε αυτή την περίπτωση αν υπάρχουν ήδη δύο στοιχεία στην λίστα τότε εξάγουμε τα δύο τελευταία στοιχεία και τα επεξεργαζόμαστε βάζοντας το operator ανάμεσα τους,κλείνοντάς τα σε παρενθέσεις και αποθηκεύοντάς το σε μία μεταβλητή.Τελικά εισάγουμε τη μεταβλητή στη τελευταία θέση της λίστας και στο τέλος του for loop εμφανίζουμε το τελευταίο στοιχείο της που στη περίπτωσή μας θα είναι η infix μορφή.

Θέμα Γ:

Αρχικά δημιουργήσαμε μία μέθοδο την isAcceptable2 που ελέγχει αν η είσοδος του χρήστη είναι έγκυρη ή όχι. Στη συνέχεια δημιουργούμε μία λίστα και εισάγουμε ένα for loop το οποίο ελέγχει κάθε στοιχείο της συμβολοσειράς που εισάγει ο χρήστης με την βοήθεια της παραπάνω μεθόδου. Αν δεν είναι έγκυρη η είσοδος τερματίζεται το πρόγραμμα αλλιώς αλλάζει κάθε στοιχείο με το συμπληρωματικό του και το εισάγουμε στη πρώτη θέση της λίστας. Με αυτό το τρόπο επιτυγχάνεται και η αντιστροφή της σειράς των στοιχείων. Στη συνέχεια εισάγουμε ένα δεύτερο for loop το οποίο ελέγχει για όλα τα στοιχεία της λίστας αν το πρώτο στοιχείο ισούται με το πρώτο στοιχείο της ακολούθείας που είχε εισάγει ο χρήστης στην αρχή. Τελος αν όλα τα στοιχεία είναι ίσα μεταξύ τους ένα προς ένα τότε το πρόγραμμα εμφανίζει ότι η ακολουθία είναι παλινδρομημένη αλλιώς εμφανίζει οτι η ακολουθία δεν είναι παλινδρομημένη.

Το πρόγραμμα επιτυγχάνει χρονική πολυπλοκότητα O(N) παρόλο που χρησιμοποιεί δύο for loops διότι είναι ανεξάρτητα μεταξύ τους οπότε η χρονική πολυπλοκότητα του προγράμματος είναι του τύπου O(N)+O(N) που ισούται με O(N).