

ΑΝΑΦΟΡΑ(ΜΕΡΟΣ Δ)

ΜΕΡΟΣ Α:

Η κλάση `StringDoubleEndedQueueImpl` κάνει `implement` την `interface StringDoubleEndedQueue` (στην οποία δηλώνονται οι μέθοδοι που υλοποιούνται στην κλάση). Αρχικά επεξεργάζεται αντικείμενα τύπου `Node` (τα οποία ορίζονται, και υλοποιούνται και οι `accessors` που χρησιμοποιούν, από την αντίστοιχη κλάση). Συγκεκριμένα στην μέθοδο `addFirst` δημιουργεί ένα τέτοιο αντικείμενο, το οποίο τοποθετείται πριν από το `head` και ύστερα γίνεται αυτό το `head` (ελέγχοντας πρώτα ότι δεν είναι άδεια η λίστα όπου σε αυτή την περίπτωση το νέο αυτό αντικείμενο είναι και το μοναδικό). Αντίστοιχα κινούμαστε και στην `addLast`. Στη μέθοδο `removeFirst`, η οποία επιστρέφει το αντικείμενο το οποίο θα διαγραφεί (εδώ είναι το πρώτο στοιχείο της λίστας), πετάει εξαίρεση σε περίπτωση που είναι άδεια αλλιώς θέτει το επόμενο στοιχείο του `head` ως `head`. (Παρόμοια και στη `removeLast`.) Οι `getFirst` και `getLast` είναι `accessors` που επιστρέφουν το πρώτο και το τελευταίο στοιχείο της λίστας αντίστοιχα. Η `printQueue` τυπώνει την λίστα σε ένα `stream` και τέλος η `size` επιστρέφει το μέγεθος της λίστας το οποίο ρυθμίζεται ανάλογα με το ποιές μέθοδοι έχουν κλειθεί για να βάλουν ή να βγάλουν στοιχεία από αυτήν (γι' αυτό και $O(1)$).

ΜΕΡΟΣ Β:

Η κλάση `PostfixToInfix` βγάζει μήνυμα στο χρήστη να εισάγει μία παράσταση σε `Postfix` μορφή, την οποία και δέχεται σε `string`. Ύστερα ελέγχεται εάν είναι σε σωστή μορφή (δηλαδή εάν τελειώνει με πράξη και εάν αποτελείται από αποδεκτούς χαρακτήρες). Αν είναι επιτρεπτή συνεχίζει με τη μετατροπή αλλιώς τυπώνει μήνυμα και ενημερώνει το χρήστη. Κατά τη μετατροπή, αντιγράφει κάθε χαρακτήρα της `string` παράστασης στην `s` λίστα που δημιουργήσαμε. Μετά ελέγχει

κάθε στοιχείο εάν είναι πράξη (μέσω της `isOperator`). Σε περίπτωση που ισχύει αποθηκεύει στο `s2` το `string` που επιστρέφεται και διαγράφεται απο το τέλος της `s`, μέσω της `removeLast`. Το ίδιο γίνεται και για το νέο τελευταίο στοιχείο της `s` αλλά στο `s1` και ύστερα εκτελείται η πράξη που ελέγχθηκε πριν ανάμεσα στα δύο αυτά `strings`. Αν όμως δεν είναι τελεστής πράξης τοποθετείται στο τέλος της `s` μέσω της `addLast`. Αυτό επαναλαμβάνεται μέχρις ότου ελεγχθεί όλη η `s` και στο τέλος τυπώνει στο χρήστη ένα μήνυμα με την `infix` μορφή της παράστασης.

ΜΕΡΟΣ Γ:

Η κλάση `DNAPalindrome` βγάζει μήνυμα στο χρήστη να εισάγει μία ακολουθία DNA την οποία αρχικά ελέγχει εάν είναι άρτιου πλήθους χαρακτήρων. Αν δεν είναι τότε σίγουρα δεν είναι και συμπληρωματικά παλίνδρομη γι' αυτό και τυπώνει μήνυμα για να ενημερώσει το χρήστη. Αν περάσει τον πρώτο έλεγχο, ξεκινά και δεύτερο μέσω της `isEnabled`, η οποία δείχνει εάν είναι οι χαρακτήρες της `string` ακολουθίας γράμματα που αντιστοιχούν σε νουκλεοτίδια. Στην περίπτωση που δεν είναι ενημερώνει το χρήστη με αντίστοιχο μήνυμα ενώ στην αντίθετη περίπτωση ελέγχει μέσω της `WatsonCrick` αν η παράσταση είναι συμπληρωματικά παλίνδρομη (που και πάλι θα ενημερώσει τον χρήστη με μήνυμα). Συγκεκριμένα όμως η μέθοδος αντιγράφει τους χαρακτήρες του `string` σε μία νέα λίστα `d` και επαναληπτικά ελέγχει μέσω της `Complemental` εάν το πρώτο και το τελευταίο στοιχείο της `d` (σε κάθε νέα επανάληψη) είναι συμπληρωματικά μεταξύ τους (σε κάθε επανάληψη αυτά τα στοιχεία αλλάζουν διότι χρησιμοποιούμε τις `removeFirst` και `removeLast` που τα διαγράφουν). Η επανάληψη θα τερματίσει αν αδειάσει όλη η `d` ή όταν βρεθεί το πρώτο ζευγάρι στοιχείων που δεν είναι συμπληρωματικά παλίνδρομο.