ΜΕΡΟΣ Α

Η διεπαφή στο πρώτο μέρος της εργασίας γίνεται με generics. Δηλαδή έχουν αντικατασταθεί τα return types και οι παράμετροι της διεπαφής και της υλοποίησης από String σε T. Επίσης, στο όνομα της κλάσης StringDoubleEndedQueueImpl αλλά και στο interface της έχει προστεθεί το σύμβολο <T> για να δηλωθεί ότι η κλάση αυτή χειρίζεται και χρησιμοποιεί generics.

Οι μέθοδοι εισαγωγής και εξαγωγής έχουν πολυπλοκότητα της τάξης O(1) δεδομένου ότι οι εντολές στο σώμα τους είναι ανεξάρτητες του μεγέθους της λίστας. Δηλαδή, όσους κόμβους και να έχει η διπλά συνδεδεμένη λίστα, ο αριθμός των εντολών και κατά συνέπεια ο χρόνος εκτέλεσης δεν επηρεάζεται.

Επιπλέον, σε κάθε μέθοδο λαμβάνεται υπ' όψιν ο σωστός χειρισμός των head, tail, next, previous node για να εξασφαλιστεί η ορθή σύνδεση των κόμβων και να μην χαθεί η λίστα.

Έχει δημιουργηθεί , ακόμα, ένα αρχείο με όνομα DLNode.java (Doubly-Linked list) το οποίο δημιουργεί αντικείμενα τύπου Node (κόμβους) που χρησιμοποιούνται στην κλαση StringDoubleEndedQueueImpl για να υλοποιηθεί η διπλά συνδεδεμένη λίστα. Το αρχείο αυτό χρησιμοποιεί generics για να μπορεί να λειτουργήσει σωστά η κλάση που καλεί τις μεθόδους του. Τα attributes της κλάσης περιέχουν τα δεδομένα του κόμβου, τους κόμβους next και previous, και την στατική μεταβλητή size (κρατάει την τιμή των προηγούμενων αντικειμένων που δημιουργούνται). Ο constructor παίρνει σαν παράμετρο το δεδομένο ενός κόμβου και το αντιστοιχεί στην private μεταβλητή item, ενώ παράλληλα αυξάνεται το size κατά 1 κάθε φορά που δημιουργείται ένας νέος κόμβος. Έπειτα, ορίζονται getters

και setters για τα δεδομένα ενός κόμβου, για τον επόμενο και τον προηγούμενο κόμβο. Τέλος, ορίζεται ένας getter για το μέγεθος της λίστας.

Έτσι, στην διεπαφή η μέθοδος size υλοποιείται με την getSize() μέσω της ουράς (το ίδιο ακριβώς θα επέστρεφε και μέσω της κεφαλής), με αποτέλεσμα η μέθοδος αυτή να έχει πολυπλοκότητα O(1).

ΜΕΡΟΣ Β

Καταρχάς, το δεύτερο ζητούμενο απαιτεί user input κι έτσι χρησιμοποιείται η κλάση Scanner του πακέτου util της java. Στη συνέχεια, ελέγχεται η έκφραση που δόθηκε από τον χρήστη μέσω μιας επιπλέον μεθόδου με όνομα validExpression η οποία επιστρέφει true αν η έκφραση είναι σε ορθή μεταθεματική μορφή (αν οι χαρακτήρες είναι είτε αριθμοί από 0 μέχρι 9 είτε κάποιος έγκυρος τελεστής και αν η έκφραση τερματίζει με τελεστή) αλλιώς επιστρέφει false.

Με τη βοήθεια της StringDoubleEndedQueueImpl δημιουργείται μια διπλά συνδεδεμένη κενή, για αρχή, λίστα με όνομα queue. Η λίστα αυτή χρησιμοποιείται για να αποθηκεύεται κάθε φορά ο σωστός χαρακτήρας για την ενθεματική μορφή της έκφρασης. Έπειτα, γίνεται προσπέλαση κάθε χαρακτήρα της δοσμένης έκφρασης και ελέγχεται αν είναι τελεστής ή τελεστέος. Αν είναι τελεστέος τότε προστίθεται στο τέλος της λίστας. Αν είναι τελεστής τότε αφαιρούνται τα δύο τελευταία στοιχεία της λίστας (η οποία θα έχει τουλάχιστον δύο στοιχεία, εφόσον η μεταθεματική μορφή προβλέπει να υπάρχουν στην αρχή της έκφρασης τουλάχιστον δύο αριθμοί και μετά τελεστής ο οποίος εφαρμόζεται στους δύο προηγούμενους αριθμούς) και ανατίθενται σε δύο strings. Με τη βοήθεια ενός άλλου string, ανατίθεται μια τιμή που αποτελείται από το πρώτο string συν τον τελεστή συν το

δεύτερο string και που περικλείεται από παρενθέσεις. Το δεύτερο αυτό string θα προστεθεί στο τέλος της λίστας. Όταν τελειώσουν όλες οι προσπελάσεις της δοσμένης έκφρασης, η λίστα θα περιέχει την έκφραση αυτή σε ενδοθεματική μορφή.

Παράδειγμα:

Έστω 56* η έκφραση σε μεταθεματική μορφή. Το πρόγραμμα θα προσπελάσει την έκφραση από τον πρώτο χαρακτήρα, δηλαδή το 5. Επειδή είναι τελεστέος θα προστεθεί κατευθείαν στο τέλος της λίστας. Το ίδιο θα συμβεί και με το 6, άρα η λίστα θα περιέχει [5,6] Ο χαρακτήρας c = * είναι τελεστής άρα το πρόγραμμα θα αφαιρέσει τις δύο τελευταίες τιμές της λίστας και θα τις αναθέσει σε δύο string op1 και op2 με αποτέλεσμα η λίστα να μην περιέχει κανένα στοιχείο. Θα έχουμε op1 = 6 και op2 = 5. Ένα άλλο string temp θα έχει τιμή (op2+c+op1) δηλαδή (5*6) το οποίο θα προστεθεί στο τέλος της λίστας. Έτσι, η λίστα θα περιέχει (5*6) το οποίο στη συνέχεια φαίνεται στην κονσόλα.

ΜΕΡΟΣ Γ

Αρχικά, χρησιμοποιείται η κλάση Scanner για το user input και στη συνέχεια ελέγχεται η ακολουθία dna που δίνεται από τον χρήστη μέσω της μεθόδου isVal που επιστρέφει true αν η ακολουθία αποτελείται από τα κεφαλαία γράμματα A, T, G, C ή με τον κενό χαρακτήρα αλλιώς επιστρέφει false.

Αν η ακολουθία έχει έναν (έγκυρο) ή κανέναν χαρακτήρα τότε προφανώς η ακολουθία είναι συμπληρωματικά παλίνδρομη, αλλιώς αντιστρέφεται η δοσμένη ακολουθία μέσω μιας συνάρτησης με όνομα reverse όπου με τη βοήθεια ενός πίνακα αντιστρέφεται το δοσμένο string. Με βάση το ανεστραμμένο string

συμπληρώνεται η διπλά συνδεδεμένη λίστα που δημιουργήθηκε. Δηλαδή, αν ο χαρακτήρας του ανεστραμμένο string είναι A στη λίστα συμπληρώνεται T, αν είναι G τότε στη λίστα συμπληρώνεται C και το αντίστροφο.

Για να συγκριθεί το δοσμένο dna με αυτό που δημιουργήθηκε με την λίστα, το πρόγραμμα φτιάχνει ένα κενό string στο οποίο προστίθεται κάθε φορά το τελευταίο στοιχείο της λίστας. Έτσι, συγκρίνονται τα δύο string και αν είναι ίσα τότε η δοσμένη ακολουθία είναι Watson-Crick complemented palindrome.