



Δομές Δεδομένων
Διδάσκουσα: Κ. Παπακωνσταντινοπούλου
Χειμερινό Εξάμηνο 2023

Εργασία 1

Ουρές: Υλοποιήσεις ΑΤΔ και εφαρμογές
Προθεσμία υποβολής: 27/11/2023, 23:59

Σκοπός της εργασίας αυτής είναι η εξοικείωση με βασικούς αφηρημένους τύπους δεδομένων όπως οι ουρές με 2 άκρα. Η εργασία αποτελείται από μία υλοποίηση ΑΤΔ (Μέρος Α) καθώς και δύο εφαρμογές (Μέρη Β και Γ). Διαβάστε προσεκτικά την εκφώνηση και τα ζητούμενα της εργασίας.

Μέρος Α [25 μονάδες]. Στο φάκελο «Εγγραφα/Εργασίες» του eclass, δίνεται η διεπαφή `StringDoubleEndedQueue` που δηλώνει τις βασικές μεθόδους για μια *ουρά*, η οποία χειρίζεται στοιχεία τύπου `String`, και στην οποία μπορεί να γίνεται εισαγωγή και διαγραφή στοιχείων και από τα 2 άκρα της. Δημιουργήστε μία υλοποίηση της διεπαφής `StringDoubleEndedQueue`.

Οδηγίες υλοποίησης:

- Η κλάση σας **πρέπει να ονομάζεται** `StringDoubleEndedQueueImpl`.
- Η υλοποίηση **πρέπει να γίνει** χρησιμοποιώντας μια *διπλά συνδεδεμένη λίστα*.
- Κάθε πράξη εισαγωγής ή εξαγωγής στοιχείου (δηλαδή κάθε εκτέλεση των μεθόδων `addFirst`, `removeFirst`, `addLast`, `removeLast`), καθώς και οι μέθοδοι `getFirst` και `getLast`, θα πρέπει να ολοκληρώνονται σε χρόνο $O(1)$, δηλαδή σε χρόνο ανεξάρτητο από το πλήθος των αντικειμένων που μπορεί να βρίσκονται μέσα στην ουρά. Η μέθοδος `size` θα πρέπει επίσης να εκτελείται σε χρόνο $O(1)$.
- Όταν η ουρά είναι άδεια, οι μέθοδοι που διαβάζουν από τη δομή θα πρέπει να πετάνε εξαίρεση τύπου `NoSuchElementException`. Η εξαίρεση `NoSuchElementException` ανήκει στην *core* βιβλιοθήκη της Java. Κάντε την `import` από το πακέτο `java.util`. Μην κατασκευάσετε δική σας εξαίρεση.
- Μπορείτε να χρησιμοποιήσετε μέρος του κώδικα που έχει παρουσιαστεί στα εργαστήρια του μαθήματος (και είναι διαθέσιμος στο eclass) ή να γράψετε εξ' ολοκλήρου τη δική σας λίστα ή να ορίσετε μόνο αντικείμενα τύπου `Node` μέσα στην κλάση της ουράς, για τους κόμβους της λίστας, χωρίς να ορίσετε κλάση λίστας. Για να αποκτήσετε καλύτερη εξοικείωση προτείνουμε να ξεκινήσετε από την αρχή και να γράψετε τις δικές σας κλάσεις (σίγουρα όμως δεν θα χάσετε μονάδες χρησιμοποιώντας ό,τι έχετε δει στο εργαστήριο).

- **Προαιρετικά:** μπορείτε να κάνετε την υλοποίηση σας με χρήση generics, ώστε να χειρίζεστε ουρές με οποιονδήποτε τύπο αντικειμένων. Υπάρχει ένα 10% bonus σε όσους χρησιμοποιήσουν generics για την εργασία. Αν το κάνετε έτσι, θα πρέπει να τροποποιήσετε κατάλληλα τον ορισμό της διεπαφής.
- **Δεν επιτρέπεται να χρησιμοποιήσετε έτοιμες υλοποιήσεις δομών τύπου λίστας, στοιβάς, ουράς, από την βιβλιοθήκη της Java** (π.χ. Vector, ArrayList κλπ). Στόχος της εργασίας είναι να εξασκηθείτε στη δημιουργία δικών σας υλοποιήσεων δομών δεδομένων.

Μέρος Β [30 μονάδες]. Γράψτε ένα πρόγραμμα-πελάτη το οποίο θα διαβάζει μία αριθμητική παράσταση σε μεταθεματική (prefix) μορφή και θα τυπώνει την αντίστοιχη ενθεματική (infix) μορφή. Για απλότητα θα θεωρήσουμε ότι οι αριθμητικές παραστάσεις περιέχουν μονοψήφιους ακέραιους αριθμούς και χρησιμοποιούν τους τελεστές +, -, *, /. Κατά τη μετατροπή στην ενθεματική παράσταση θα πρέπει να προσθέσετε και τις παρενθέσεις όπου χρειάζεται. Θα πρέπει επίσης να τυπώνετε κάποιο μήνυμα λάθους αν η παράσταση δεν είναι σε έγκυρη προθεματική μορφή (π.χ. αν περιέχει και άλλα σύμβολα εκτός των αριθμών και των τελεστών, ή αν δεν ξεκινάει με τελεστή).

Οδηγίες υλοποίησης:

- Το πρόγραμμά σας **πρέπει να λέγεται** PrefixToInfix.java.
- Θα πρέπει να χρησιμοποιήσετε την υλοποίηση της ουράς από το Μέρος Α.
- Το πρόγραμμά σας θα ζητάει από το χρήστη να πληκτρολογήσει την παράσταση σε προθεματική μορφή, την οποία θα διαβάζετε από το standard input. Για την ανάγνωση της ακολουθίας μπορείτε να χρησιμοποιήσετε τη βοηθητική κλάση Scanner και να αποθηκεύσετε έτσι την είσοδο σε ένα String ή να χρησιμοποιήσετε όποιο άλλο τρόπο σας βολεύει για να διαβάσετε χαρακτήρες (π.χ. αποθήκευση σε πίνακα χαρακτήρων). Από εκεί και μετά όμως, η επεξεργασία θα γίνει με τη χρήση της ουράς για να κάνετε την μετατροπή σε ενθεματική μορφή.

Μέρος Γ [35 μονάδες]. Γράψτε ένα πρόγραμμα-πελάτη για να λύσετε το παρακάτω πρόβλημα χρησιμοποιώντας ξανά την υλοποίηση από το Μέρος Α.

Μία ακολουθία DNA είναι ένα αλφαριθμητικό (String) που αποτελείται από τα γράμματα A, T, C, G. Τα γράμματα αυτά αντιστοιχούν στα νουκλεοτίδια Αδενίνη, Θυμίνη, Κυτοσίνη, Γουανίνη. Τα νουκλεοτίδια A και T είναι συμπληρωματικά μεταξύ τους και θα λέμε ότι το A είναι το συμπλήρωμα του T και αντιστρόφως. Ομοίως το C είναι το συμπλήρωμα του G και το G το συμπλήρωμα του C. Μία ακολουθία DNA είναι *Watson-Crick complemented palindrome* (συμπληρωματικά παλίνδρομη) αν, **όταν αντικαταστήσουμε κάθε νουκλεοτίδιο με το συμπλήρωμά του και μετά αντιστρέψουμε την σειρά, τότε παίρνουμε την ίδια ακολουθία με την αρχική.** Για παράδειγμα το string AAAACGTTTT είναι συμπληρωματικά παλίνδρομο. Αντιθέτως το ATAATA δεν είναι (προσέξτε ότι η έννοια αυτή είναι διαφορετική από το να είναι απλά παλίνδρομη μια ακολουθία χαρακτήρων). Οι ακολουθίες DNA με αυτή την ιδιότητα έχουν σημαντικούς βιολογικούς ρόλους. Για παράδειγμα τα καρκινικά κύτταρα συχνά μεγαλώνουν τα γονίδιά τους δημιουργώντας τέτοιου είδους συμπληρωματικά παλίνδρομα DNA.

Γράψτε ένα πρόγραμμα το οποίο διαβάζει μια ακολουθία χαρακτήρων που αναπαριστά DNA ακολουθία και αποφασίζει αν είναι Watson-Crick complemented palindrome. Η main σας θα πρέπει να διαβάζει την ακολουθία DNA από το standard input, να ελέγχει αν είναι έγκυρη (π.χ. δεν περιέχει κενά ενδιάμεσα ή μη αποδεκτούς χαρακτήρες, μη κεφαλαία γράμματα), και στο τέλος να εκτυπώνει το σχετικό αποτέλεσμα.

Επιπλέον παραδείγματα από Watson-Crick παλινδρόμηση:

Ναι: "ΑΤΑΤΑΤΑΤ", και "" (το κενό string ικανοποιεί την ιδιότητα).
Όχι: "ΑΑΑΑ", "ΑΑΑΑΓΤΤΤΤ", "ΑΤΑΑΤΑ", "ΖΖΖΖ" και "ΑαΤΤ".

Οδηγίες υλοποίησης:

- Το πρόγραμμα σας **πρέπει να ονομάζεται** DNAPalindrome.java.
- Η υλοποίηση σας θα χρησιμοποιεί μόνο 1 αντικείμενο ουράς με δύο άκρα από το Μέρος Α.
- **Απαιτήσεις πολυπλοκότητας [15/35 μονάδες του Μέρους Γ]:** Αν διαβάσετε μια ακολουθία DNA με N νουκλεοτίδια, το πρόγραμμα σας πρέπει να εκτελείται σε χρόνο $O(N)$. Επιπλέον, μετά το διάβασμα της εισόδου, η επεξεργασία της ουράς που θα χρησιμοποιήσετε θα πρέπει να γίνει με 1 loop. **Εκμεταλλευτείτε** τις δυνατότητες που σας δίνει η ουρά 2 άκρων (μπορεί να λυθεί και με λιγότερο αποδοτικό τρόπο, αλλά τότε θα χάσετε κάποιες μονάδες).

Μέρος Δ - Αναφορά παράδοσης [10 μονάδες]. Ετοιμάστε μία σύντομη αναφορά σε pdf αρχείο (μην παραδώσετε docx ή txt ή rtf αρχεία!) με όνομα project1-report.pdf (ενδεικτικά όρια: τουλάχιστον 2 και το πολύ 5 σελίδες), στην οποία θα αναφερθείτε στα εξής:

- Εξηγήστε συνοπτικά πώς υλοποιήσατε την διεπαφή στο μέρος Α, π.χ. γιατί είναι σε $O(1)$ οι λειτουργίες εισαγωγής/εξαγωγής και η μέθοδος size (άνω όριο 1 σελίδα).
- Για το μέρος Β, εξηγήστε πώς χρησιμοποιήσατε την υλοποίηση από το μέρος Α για να κάνετε την μετατροπή σε ενθεματική μορφή. Μπορείτε να συμπεριλάβετε και κάποιο απλό παράδειγμα για να περιγράψετε την ιδέα της μετατροπής (άνω όριο 2 σελίδες).
- Ομοίως για το Μέρος Γ, εξηγήστε την ιδέα σας για να λύσετε το ζητούμενο πρόβλημα (άνω όριο 2 σελίδες).
- Μπορείτε επίσης να προσθέσετε οποιαδήποτε πληροφορία κρίνεται απαραίτητη για να γνωρίζουν οι βοηθοί του μαθήματος πώς εκτελείται ο κώδικάς σας.

Οδηγίες Παράδοσης

Η εργασία σας θα πρέπει να μην έχει συντακτικά λάθη και να μπορεί να μεταγλωττίζεται. **Εργασίες που δεν μεταγλωττίζονται χάνουν το 50% της συνολικής αξίας. Η διόρθωση των εργασιών από τους βοηθούς του μαθήματος αφορά την αξιολόγηση μεταγλωττίσιμων υλοποιήσεων και όχι την εύρεση λαθών μεταγλώττισης.**

Τα παραδοτέα της εργασίας είναι τα εξής:

- Τον πηγαίο κώδικα (source code). Τοποθετήστε σε ένα φάκελο με όνομα **src** τα αρχεία java που έχετε φτιάξει. Χρησιμοποιήστε τα ονόματα των κλάσεων όπως ακριβώς δίνονται στην εκφώνηση. Επιπλέον, φροντίστε να συμπεριλάβετε όποια άλλα αρχεία πηγαίου κώδικα απαιτούνται για να μεταγλωττίζεται η εργασία σας. Φροντίστε επίσης να προσθέσετε επεξηγηματικά σχόλια όπου κρίνεται απαραίτητο στον κώδικά σας.
- Την αναφορά παράδοσης.

Όλα τα παραπάνω αρχεία θα πρέπει να μουν σε ένα αρχείο zip. Το όνομα που θα δώσετε στο αρχείο αυτό θα είναι ο αριθμός μητρώου σας πχ. 3130056_3130066.zip ή 3130056.zip (αν δεν είστε σε ομάδα). Στη συνέχεια θα υποβάλετε το zip αρχείο σας στην περιοχή του μαθήματος «Εργασίες» στο e-class. Δεν χρειάζεται υποβολή και από τους 2 φοιτητές μιας ομάδας, αρκεί να υποβάλει μόνο ο ένας.

Η προθεσμία υποβολής της εργασίας είναι Δευτέρα 27 Νοεμβρίου 2023 και ώρα 23:59.