

Άσκηση 3

Καταληκτική ημερομηνία και ώρα ηλεκτρονικής υποβολής: 19/7/2020, 23:59:59

Επιστροφή στο σπίτι, ξανά (0.25 βαθμοί)

Το πρόβλημα με την επιστροφή του Σωτήρη στο σπίτι είναι γνωστό από την προηγούμενη σειρά ασκήσεων της φετινής χρονιάς. Το ζητούμενο αυτής της άσκησης είναι να γραφεί η λύση του σε Java. Το πρόγραμμά σας θα πρέπει να έχει την ίδια συμπεριφορά με τα προγράμματα σε Python που παραδώσατε για τη δεύτερη σειρά ασκήσεων. Για τα παραδείγματα της εκφώνησης της δεύτερης σειράς, η έξοδος του προγράμματός σας πρέπει να είναι η εξής (προσέξτε ότι αυτό σημαίνει πως η κύρια κλάση σας —αυτή με τη μέθοδο `main`— πρέπει να ονομάζεται `StayHome`, προσοχή στα μικρά και τα κεφαλαία γράμματα):

```
$ java StayHome s1.txt
15
DDRRRRRRDRRRDR

$ java StayHome s2.txt
12
UUURRRRRDRDDR

$ java StayHome s3.txt
IMPOSSIBLE
```

Βρείτε το εμβόλιο (0.25+0.25+0.25 = 0.75 βαθμοί)

Αφού τη γλύτωσε ο Σωτήρης και γύρισε στο σπίτι του, μας ζητά για μια τελευταία φορά για φέτος να τον βοηθήσουμε. Έχει καταφέρει να απομονώσει το RNA του κορωνοϊού, το οποίο όπως ίσως θυμόμαστε από τη βιολογία είναι μια αλυσίδα αζωτούχων βάσεων: “A”, “C”, “G” και “U”. Έχουμε λοιπόν μια συμβολοσειρά αποτελούμενη από χαρακτήρες αυτού του αλφαβήτου που περιγράφει το RNA του ιού.¹

Ο Σωτήρης θέλει να κατασκευάσει ένα εμβόλιο που θα σκοτώνει τον ιό. Για να γίνει αυτό, πρέπει το RNA του ιού να μετασχηματιστεί έτσι ώστε όλες οι ίδιες βάσεις να εμφανίζονται συνεχόμενες στην αλυσίδα. Με άλλα λόγια, όλα τα “A” πρέπει να είναι μαζί, όλα τα “C” πρέπει να είναι μαζί, κ.ο.κ. Η σειρά με την οποία θα εμφανίζονται τα groups των βάσεων δεν έχει σημασία.

Το RNA μετασχηματίζεται με μια πολύπλοκη διαδικασία που ο Σωτήρης μάς την εξήγησε κι εμείς την καταλάβαμε ως εξής. Τοποθετούμε τους χαρακτήρες που περιγράφουν την αλυσίδα των βάσεων σε μία στοίβα. Υπάρχει και μία δεύτερη, αρχικά κενή στοίβα, στην οποία στο τέλος θα βρεθεί το αποτέλεσμα του μετασχηματισμού. Έχουμε στη διάθεσή μας τρεις κινήσεις:

- **push** (“p”): αφαιρεί τη βάση που βρίσκεται στην κορυφή της πρώτης στοίβας και την τοποθετεί στην κορυφή της δεύτερης στοίβας.
- **complement** (“c”): αντικαθιστά κάθε βάση της πρώτης στοίβας με τη «συμπληρωματική» της — τα συμπληρωματικά ζεύγη βάσεων είναι “A-U” και “C-G”.
- **reverse** (“r”): αντιστρέφει το περιεχόμενο της δεύτερης στοίβας, έτσι ώστε η κορυφή της να βρεθεί στον πυθμένα και αντίστροφα.

¹ Μέχρι εδώ περίπου δεν πρέπει να έχουμε γράψει ιδιαίτερες κοτσάνες, όλα τα επόμενα όμως μάλλον αντιβαίνουν τη Βιολογία όπως τουλάχιστον τη γνωρίζουμε σήμερα.

Για παράδειγμα, αν το RNA του ιού που θα τοποθετήσουμε αρχικά στην πρώτη στοίβα είναι “GUACA” και εφαρμόσουμε την ακολουθία κινήσεων “rrrgccrrr”, το αποτέλεσμα θα είναι “CCAAA” και αυτό έχει την επιθυμητή μορφή (όλα τα “A” είναι μαζί όπως και όλα τα “C”). Τα ενδιάμεσα περιεχόμενα των δύο στοίβων μετά από κάθε κίνηση είναι τα εξής (οι πρώτη στοίβα έχει την κορυφή της στα δεξιά ενώ η δεύτερη στοίβα έχει την κορυφή της στα αριστερά):

Κίνηση	Πρώτη στοίβα	Δεύτερη στοίβα
	GUACA	
p	GUAC	A
p	GUA	CA
r	GUA	AC
p	GU	AAC
c	CA	AAC
p	C	AAAC
r	C	CAAA
p		C AAAA

Τα στοιχεία εισόδου θα διαβάζονται από ένα αρχείο με μορφή σαν και αυτή που φαίνεται στα παραδείγματα παρακάτω. Η πρώτη γραμμή του αρχείου θα περιέχει έναν ακέραιο αριθμό **N** ($1 \leq N \leq 10$), το πλήθος των ερωτημάτων που θα ακολουθήσουν. Κάθε μία από τις επόμενες **N** γραμμές θα περιέχει μία συμβολοσειρά αποτελούμενη από έναν ή περισσότερους χαρακτήρες του αλφαβήτου "ACGU".

Περιορισμοί: Το μήκος των συμβολοσειρών δε θα υπερβαίνει το 100 και οι περιπτώσεις ελέγχου θα είναι τέτοιες ώστε ένας σχετικά απλός BFS solver (σαν αυτόν που είδαμε στο εργαστήριο της Java) να μπορεί να τις λύσει εντός των ορίων χρόνου και μνήμης.

Παρακάτω δίνεται ένα παράδειγμα σε Python, Java και Prolog.

Σε Python

```
$ python3 vaccine.py vl.txt
pp
pprpp
ppcpp
pprppp
pprprprppp
pcpcpcpcpcpp
pprprpcrprprprprprprprprprpr
ppcpcpcprprpprppcpcpprppcprp
```

Σε Java

```
$ java Vaccine v1.txt  
pp  
pprpp  
ppcpcp  
pprrppp  
pprrprprppp  
pcpcpcpcpcpp  
pprpccrprprprprprprprprprpr  
ppccpcpcpcpcpcpcpcpcpcpcpcpcpc
```

$\Sigma\varepsilon$ Prolog

```
?- vaccine('v1.txt', Answer).    % will be printed without a line break below
Answer = [pp, pprpp, ppcpp, prrppp, prrrppppp, pcpcpcpcpcpp,
prprcpcrcprprprprprprprprpr, ppcpcpcpprrpppcpcpprrppcprp] ;
false.
```

```
pp
pprpp
ppcpp
pprppp
pprprprppp
pcpcpcpcpcpp
pprprpcrprprprprprprprprprpr
pcpcpcpprpprpppcpcpprpppcprp
false.
```

όπου το αρχείο με τα δεδομένα εισόδου είναι το εξής (η εντολή `cat` είναι εντολή του Unix):

```
$ cat vl.txt
8
GG
UAGA
GAGA
GUACA
UGACACA
CACGCGC
AGAGUGUCUGUCU
GAUCCGCCUGCACGCC
```

Περαιτέρω οδηγίες για τις ασκήσεις

- Μπορείτε να δουλέψετε σε ομάδες το πολύ δύο ατόμων. Μπορείτε αν θέλετε να σχηματίσετε διαφορετική ομάδα σε σχέση με την προηγούμενη σειρά ασκήσεων – οι ομάδες στο σύστημα υποβολής είναι έτσι και αλλιώς καινούργιες για κάθε σειρά ασκήσεων.
- Δεν επιτρέπεται να μοιράζεστε τα προγράμματά σας με συμφοιτητές εκτός της ομάδας σας ή να τα βάλετε σε μέρος που άλλοι μπορούν να τα βρουν (π.χ. σε κάποια σελίδα στο διαδίκτυο, σε ιστοσελίδες συζητήσεων, ...). Σε περίπτωση που παρατηρηθούν «περιέργες» ομοιότητες σε προγράμματα, ο βαθμός των εμπλεκόμενων φοιτητών σε *όλες τις σειρές ασκήσεων* γίνεται αυτόματα μηδέν ανεξάρτητα από το ποια ομάδα... «εμπνεύστηκε» από την άλλη.
- Μπορείτε να χρησιμοποιήσετε «βοηθητικό» κώδικα (π.χ. κάποιο κώδικα που διαχειρίζεται κάποια δομή δεδομένων) που βρήκατε στο διαδίκτυο στα προγράμματά σας, με την προϋπόθεση ότι το πρόγραμμά σας περιέχει σε σχόλια την παραδοχή για την προέλευση αυτού του κώδικα και ένα σύνδεσμο σε αυτόν.
- Τα προγράμματα σε Python πρέπει να είναι σε ένα αρχείο και να δουλεύουν σε Python 3.7.3. (Προσέξτε ότι η Python 2 είναι διαφορετική διάλεκτος της Python!)
- Τα προγράμματα σε Prolog πρέπει να είναι σε ένα αρχείο και να δουλεύουν σε κάποιο από τα παρακάτω συστήματα SWI Prolog (8.0.2), GNU Prolog (1.3.0) ή YAP (6.2.2).
- Ο κώδικας των προγραμμάτων σε Java μπορεί να βρίσκεται σε περισσότερα του ενός αρχείου αν θέλετε, αλλά θα πρέπει να μπορεί να μεταγλωττιστεί χωρίς προβλήματα με τον Java compiler με εντολές της μορφής: `javac StayHome.java` και `javac Vaccine.java`. Μην ορίσετε δικά σας packages! Η υποβολή σας σε Java μπορεί είτε να είναι ένα μόνο `.java` αρχείο ή να αποτελείται από ένα `.zip` αρχείο ενός directory το οποίο περιέχει τα `.java` αρχεία της υποβολής σας (και μόνο αυτά – μην υποβάλλετε `.class` αρχεία). Σε κάθε περίπτωση, η υποβολή σας πρέπει να έχει ένα αρχείο με τα ονόματα που φαίνονται παραπάνω σε αυτήν την παράγραφο.
- Η υποβολή των προγραμμάτων θα γίνει ηλεκτρονικά μέσω του moodle, όπως και στην προηγούμενη άσκηση, και για να μπορέσετε να τις υποβάλλετε, τα μέλη της ομάδας σας (και οι δύο) θα πρέπει να έχουν ήδη λογαριασμό στο moodle. Θα υπάρξει σχετική ανακοίνωση μόλις το σύστημα υποβολής καταστεί ενεργό. Τα προγράμματά σας πρέπει να διαβάζουν την είσοδο όπως αναφέρεται και δεν πρέπει να έχουν κάποιου άλλους είδους έξοδο εκτός από τη ζητούμενη διότι δε θα γίνουν δεκτά από το σύστημα υποβολής.