

## Άσκηση 3

Καταληκτική ημερομηνία και ώρα ηλεκτρονικής υποβολής: 21/7/2019, 23:59:59

### Σώσε τη γάτα, ξανά (0.25 βαθμοί)

Το πρόβλημα με τη σωτηρία της γάτας είναι κι αυτό γνωστό από τις δύο πρώτες σειρές 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 C/C++ που παραδώσατε για την πρώτη σειρά ασκήσεων. Για τα παραδείγματα της εκφώνησης της πρώτης 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 “`SaveTheCat`”, προσοχή στα μικρά και τα κεφαλαία γράμματα):

```
$ java SaveTheCat a1.txt          $ java SaveTheCat a3.txt
infinity                          15
stay                               DDDRRRURR

$ java SaveTheCat a2.txt          $ java SaveTheCat a4.txt
5                                  infinity
RDDLL                             LLUU
```

### Η τρελομηχανή Ztalloc (0.25+0.25+0.25 = 0.75 βαθμοί)

Μία πολύ μυστήρια και ευτυχώς υποθετική υπολογιστική μηχανή διαθέτει μόνο έναν καταχωρητή, που χωράει ένα φυσικό αριθμό το πολύ εξαψήφιο (0 έως 999.999). Η μηχανή αυτή υποστηρίζει μόνο δύο πράξεις:

- `half` (“h”), που διαιρεί το περιεχόμενο του καταχωρητή δια δύο (ακέραια διαίρεση), και
- `triple plus one` (“t”), που πολλαπλασιάζει το περιεχόμενο του καταχωρητή επί τρία και προσθέτει ένα. Η πράξη αυτή προκαλεί σφάλμα εκτέλεσης αν το αποτέλεσμα έχει περισσότερα από έξι ψηφία και το πρόγραμμα που την εκτελεί δεν είναι έγκυρο.

Ένα πρόγραμμα αυτής της μηχανής είναι μία (πιθανώς κενή) ακολουθία από χαρακτήρες “h” ή/και “t”, που αντιστοιχούν στις παραπάνω δύο εντολές και εκτελούνται κατά σειρά από αριστερά προς τα δεξιά. Το «αποτέλεσμα» της εκτέλεσης του προγράμματος είναι το τελικό περιεχόμενο του καταχωρητή και, προφανώς, εξαρτάται από το αρχικό περιεχόμενο του καταχωρητή, το οποίο θα ονομάζουμε «είσοδο».

Για παράδειγμα, αν η είσοδος είναι 0 και εκτελέσουμε το πρόγραμμα “`ttththtth`”, το αποτέλεσμα θα είναι ο αριθμός 42, αφού το περιεχόμενο του καταχωρητή της μηχανής θα γίνει διαδοχικά:  $0 \rightarrow 1 \rightarrow 4 \rightarrow 13 \rightarrow 6 \rightarrow 19 \rightarrow 9 \rightarrow 28 \rightarrow 85 \rightarrow 42$ .

Δυστυχώς δε γνωρίζουμε ποια ακριβώς είναι η τιμή της εισόδου, γνωρίζουμε μόνο ότι βρίσκεται σε ένα κλειστό διάστημα φυσικών αριθμών  $[L_{in}, R_{in}]$ . Θέλουμε να εξασφαλίσουμε ότι το αποτέλεσμα θα βρίσκεται οπωσδήποτε στο κλειστό διάστημα  $[L_{out}, R_{out}]$ . Βρείτε το ελάχιστο δυνατό έγκυρο πρόγραμμα που επιτυγχάνει το ζητούμενο, δηλαδή το πρόγραμμα που έχει τις λιγότερες δυνατές πράξεις. Αν υπάρχουν περισσότερα έγκυρα προγράμματα που επιτυγχάνουν το ζητούμενο με το ίδιο πλήθος πράξεων, τότε βρείτε το λεξικογραφικά μικρότερο.

Αυτό που ζητάει η άσκηση είναι να γραφούν τρία προγράμματα (ένα σε Python, ένα σε Prolog και ένα σε Java) τα οποία να παίρνουν ως είσοδο μία ακολουθία από ερωτήματα της παραπάνω

μορφής (δηλαδή τετράδες  $L_{in}, R_{in}, L_{out}, R_{out}$ ) και για κάθε μία υπολογίζουν το ελάχιστο ζητούμενο πρόγραμμα.

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  $Q$ , το πλήθος των ερωτημάτων που θα ακολουθήσουν. Κάθε μία από τις επόμενες  $Q$  γραμμές θα περιέχει τέσσερις ακέραιους αριθμούς  $L_{in}, R_{in}, L_{out}, R_{out}$ , χωρισμένους ανά δύο με ένα κενό διάστημα.

Το πρόγραμμά σας θα πρέπει να εκτυπώνει στην τυπική έξοδο (standard output) συνολικά  $Q$  γραμμές, κάθε μία από τις οποίες θα περιέχει το ελάχιστο πρόγραμμα για το αντίστοιχο ερώτημα. Αν για κάποιο ερώτημα η απάντηση είναι το κενό πρόγραμμα, το πρόγραμμά σας θα πρέπει να τυπώνει στην αντίστοιχη γραμμή τη συμβολοσειρά "EMPTY". Αν για κάποιο ερώτημα δεν υπάρχει απάντηση (δηλαδή κανένα πρόγραμμα δεν εξασφαλίζει ότι το αποτέλεσμα θα είναι στο επιθυμητό διάστημα) τότε το πρόγραμμά σας θα πρέπει να εκτυπώνει στην αντίστοιχη γραμμή τη συμβολοσειρά "IMPOSSIBLE".

### Περιορισμοί:

- Για το 30% της βαθμολογίας θα είναι:  $1 \leq Q \leq 10$  και  $0 \leq L_{in} = R_{in} \leq 1.000.000$ .
- Για το υπόλοιπο 70% της βαθμολογίας θα είναι  $1 \leq Q \leq 10$ ,  $0 \leq L_{in} < R_{in} \leq 1.000.000$  και το συνολικό πλήθος των καταστάσεων που θα χρειάζεται να επισκεφθεί ένας απλός BFS solver (σαν αυτόν που είδαμε στο εργαστήριο της Java) δε θα υπερβαίνει το 1.000.000.
- Για ένα bonus 40% της βαθμολογίας θα είναι  $1 \leq Q \leq 10$  και  $0 \leq L_{in} \leq R_{in} \leq 1.000.000$  χωρίς κανέναν άλλο περιορισμό. (Ο διδάσκων εκτιμά ότι αυτό θα είναι αρκετά δύσκολο να το πετύχετε, ειδικά σε Python και σε Prolog.)

Παρακάτω δίνονται κάποια παραδείγματα σε Python, Prolog και Java. Προσέξτε ότι το δεύτερο παράδειγμα (z2.txt) εμπίπτει στην κατηγορία bonus.

#### Σε Python

```
$ python3 ztalloc.py z1.txt
tthtttht
ttthttth
htttthttth
EMPTY
hhhtthttttttttttt
ttttthttthttthttttt
hhhtthttttttttttt
hhhhtthttthttttttttt
IMPOSSIBLE
hhtthttthttttttttt

$ python3 ztalloc.py z2.txt
htthttthttthttthttthttthttthttth
IMPOSSIBLE
```

#### Σε Java

```
$ java Ztalloc z1.txt
tthtttht
ttthttth
htttthttth
EMPTY
hhhtthttttttttttt
ttttthttthttthttttt
hhhtthttttttttttt
hhhhtthttthttttttttt
IMPOSSIBLE
hhtthttthttttttttt

$ java Ztalloc z2.txt
htthttthttthttthttthttthttthttth
IMPOSSIBLE
```

#### Σε Prolog

```
?- ztalloc('z1.txt', Answer).
Answer = [tthtttht, ttthttth, httthttth, 'EMPTY'|...].

?- ztalloc('z1.txt', Answer), maplist(writeln, Answer).
tthtttht
ttthttth
htttthttth
EMPTY
hhhtthttttttttttt
ttttthttthttthttttt
hhhtthttttttttttt
hhhhtthttthttttttttt
IMPOSSIBLE
hhtthttthttttttttt
Answer = [tthtttht, ttthttth, httthttth, 'EMPTY'|...].
```

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

```
$ cat z1.txt
10
0 0 17 17
0 0 42 42
0 1 42 42
18 20 17 42
2 5 478958 729952
0 0 999888 999999
17 17 888999 999888
0 9 888999 999888
0 0 897215 897216
1 2 892871 918276

$ cat z2.txt
2
0 1 466607 466608
0 1 897215 897216
```

## 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 3.4.2. (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 (6.6.6), 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 SaveTheCat.java` και `javac Ztalloc.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 υποβάλετε `.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.