

Άσκηση 2

Καταληκτική ημερομηνία και ώρα ηλεκτρονικής υποβολής: 18/6/2017, 23:59:59

Τα ωραία χωριά ωραία ενώνονται (0.25+0.25 = 0.5 βαθμοί)

Μετά το 144^ο μνημόνιο, οι περισσότεροι δρόμοι που συνέδεαν τα χωριά της Ελλάδας μεταξύ τους έχουν καταστραφεί. Η χώρα έχει N χωριά, αριθμημένα από 1 έως και N , και M βατούς δρόμους διπλής κατεύθυνσης που καθένας συνδέει δύο διαφορετικά χωριά. Δεν υπάρχουν περισσότεροι από ένας δρόμοι με άκρα στα ίδια χωριά. Αυτοί οι δρόμοι δεν εξασφαλίζουν την δυνατότητα οδικής σύνδεσης δυο οποιωνδήποτε χωριών. Αντίθετα, τα χωριά σχηματίζουν ομάδες, που σε κάθε ομάδα δύο οποιαδήποτε χωριά συνδέονται οδικώς μεταξύ τους. Όμως, δεν υπάρχει οδική σύνδεση ανάμεσα σε χωριά που ανήκουν σε διαφορετικές ομάδες.



spoiler!

Το κράτος, με σύνθημα «έρχεται η επανασύνδεση», σχεδιάζει την ανασυγκρότηση του οδικού δικτύου. Όμως, λόγω περιορισμένων πόρων και επειδή την τελευταία φορά που «βγήκε στις αγορές» άρπαξε ένα πολύ γερό κρουολόγημα και είναι ακόμα με αντιβίωση, έχει την δυνατότητα κατασκευής το πολύ K νέων δρόμων, καθένας από τους οποίους θα συνδέει δυο χωριά για τα οποία δεν υπάρχει οδική σύνδεση. Μετά την κατασκευή αυτών των νέων δρόμων, αν έχει γίνει σωστή σχεδίαση, το πλήθος των ομάδων χωριών θα μειωθεί γιατί περισσότερα χωριά θα συνδέονται οδικώς μεταξύ τους.

Η άσκηση σας ζητάει να γράψετε δύο προγράμματα (ένα σε ML και ένα σε Java) τα οποία να βρίσκουν το ελάχιστο δυνατό πλήθος ομάδων που μπορεί να μείνει μετά την κατασκευή των νέων δρόμων.

Τα στοιχεία εισόδου θα διαβάζονται από ένα αρχείο όπως φαίνεται στα παραδείγματα που ακολουθούν. Η πρώτη γραμμή του περιέχει τρεις ακέραιους αριθμούς N , M , K , χωρισμένους ανά δύο με ένα κενό διάστημα: το πλήθος N ($1 \leq N \leq 1.000.000$) των χωριών, το πλήθος M ($1 \leq M \leq 2.000.000$) των υπαρχόντων δρόμων, και το πλήθος K ($0 \leq K \leq 1.000.000$) των νέων δρόμων που πρόκειται να κατασκευαστούν. Κάθε μία από τις επόμενες M γραμμές περιέχει δύο ακέραιους αριθμούς A και B , χωρισμένους με ένα κενό διάστημα, όπου $1 \leq A, B \leq N$ και $A \neq B$. Αυτό σημαίνει ότι υπάρχει ένας βατός δρόμος που συνδέει τα χωριά A και B .

Παρακάτω δίνονται κάποια παραδείγματα σε ML και σε Java.

Σε MLton, ή σε OCaml	Σε SML/NJ	Σε Java
<pre>\$./villages v1.txt 3</pre>	<pre>- villages "v1.txt"; val it = 3 : int</pre>	<pre>\$ java Villages v1.txt 3</pre>
<pre>\$./villages v2.txt 1</pre>	<pre>- villages "v2.txt"; val it = 1 : int</pre>	<pre>\$ java Villages v2.txt 1</pre>
<pre>\$./villages v3.txt 1</pre>	<pre>- villages "v3.txt"; val it = 1 : int</pre>	<pre>\$ java Villages v3.txt 1</pre>

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

```
$ cat v1.txt
7 2 2
1 2
6 5
```

```
$ cat v2.txt
4 2 3
1 2
4 3
```

```
$ cat v3.txt
4 3 0
3 2
1 4
1 3
```

Για το 1ο παράδειγμα, αρχικά υπάρχουν πέντε ομάδες χωριών: {1, 2}, {3}, {4}, {5, 6} και {7}. Αν κατασκευαστούν δύο νέοι δρόμοι, π.χ. μεταξύ των χωριών (3, 7) και των χωριών (3, 4), τότε θα προκύψουν μόνο τρεις ομάδες: {1, 2}, {3, 4, 7} και {5, 6}. Αυτός είναι και ο ελάχιστος αριθμός ομάδων που μπορούν να προκύψουν μετά την κατασκευή δύο δρόμων. Στο 2ο παράδειγμα, αρκεί να κατασκευαστεί ένας νέος δρόμος για να μείνει μόνο μία ομάδα που να περιέχει όλα τα χωριά. Στο 3ο παράδειγμα, υπάρχει εξ αρχής μόνο μία ομάδα χωριών και αυτή θα παραμείνει, παρόλο που το κράτος δεν μπορεί να κατασκευάσει κανέναν δρόμο ($K=0$).

Ο Λάκης αντεπιτίθεται (0.25+0.25 = 0.5 βαθμοί)

Θυμάστε τον Λάκη τον εξωγήινο, που δουλεύει ντελιβεράς στο εστιατόριο στην άκρη του σύμπαντος; Αν όχι, ξαναδιαβάστε την πρώτη άσκηση.

Έχουν περάσει μερικά χρόνια και διάφορα πράγματα έχουν αλλάξει στο σύμπαν, για να το θέσουμε κομψά. Κατ' αρχάς έχουν εξαφανιστεί όλες οι σκουληκότρυπες. Επίσης, έχει αρχίσει να φυσάει ένας δαιμονισμένος κοσμικός άνεμος από τη βορειοδυτική γωνία του σύμπαντος (πάνω-αριστερά στο χάρτη) προς τη νοτιοανατολική (κάτω-δεξιά). Ως αποτέλεσμα αυτού:

Οι κινήσεις "R" και "D" κοστίζουν 1 λεπτό, γιατί ευνοούνται από τον άνεμο.

Οι κινήσεις "L" και "U" κοστίζουν 2 και 3 λεπτά, αντίστοιχα, γιατί είναι κόντρα στον άνεμο.

Η άσκηση σας ζητάει να γράψετε δύο προγράμματα (ένα σε ML και ένα σε Java) τα οποία βρίσκουν μία ελάχιστου κόστους ακολουθία κινήσεων για να παραδώσει ο Λάκης την παραγγελία του, όπως στην προηγούμενη άσκηση. Αν υπάρχουν περισσότερες διαφορετικές ακολουθίες με ελάχιστο κόστος, αρκεί και πάλι να βρείτε μία (οποιαδήποτε) από αυτές.

Η είσοδος του προγράμματός σας διαβάζεται από ένα αρχείο αποτελούμενο από N γραμμές, κάθε μία από τις οποίες περιέχει M σύμβολα. Το αρχείο αυτό αναπαριστά το χάρτη.

Παρακάτω δίνονται κάποια παραδείγματα σε C/C++ και σε ML.

Σε MLton, ή σε OCaml

```
$ ./moredeli map1.txt
16 URRRDDLLU

$ ./moredeli map2.txt
26 DLLLLUUUURDRD
```

Σε Java

```
$ java Moredeli map1.txt
16 URRRDDLLU

$ java Moredeli map2.txt
26 DLLLLUUUURDRD
```

Σε SML/NJ

```
- moredeli "map1.txt";
val it = (16,"URRRDDLLU") : int * string

- moredeli "map2.txt";
val it = (26,"DLLLLUUUURDRD") : int * string
```

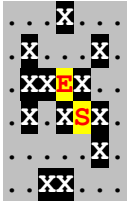
όπου τα αρχεία εισόδου είναι τα εξής (η εντολή `cat` είναι όπως είπαμε εντολή του Unix):

```
$ cat map1.txt
```



spoiler

```
$ cat map2.txt
```



Στο 1ο παράδειγμα, ο Λάκης δεν έχει πολλά να σκεφτεί. Στο 2ο παράδειγμα, όμως, ο Λάκης πρέπει να αποφασίσει αν θα κάνει τον κύκλο από δεξιά ή από αριστερά. Μετρώντας προσεκτικά, βλέπει ότι πηγαίνοντας από δεξιά (“DDRRUUUUULLDL”) θα κάνει συνολικά 27 λεπτά, ενώ πηγαίνοντας από αριστερά (“DLLLLUUUUURDR”) θα κάνει 26 λεπτά.

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 μοιράζετε 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 «περίεργες» 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 ML πρέπει να είναι σε ένα αρχείο και να δουλεύουν σε SML/NJ v110.76 ή σε MLton 20100608 ή σε Objective Caml version 4.01.0. Το σύστημα ηλεκτρονικής υποβολής επιτρέπει να επιλέξετε μεταξύ αυτών των υλοποιήσεων της ML.
- 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 Villages.java` και `javac Moredeli.java`. 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 άλλου είδους έξοδο εκτός από τη ζητούμενη διότι δε θα γίνουν δεκτά από το σύστημα υποβολής.