



Γλώσσες Προγραμματισμού II

Αν δεν αναφέρεται διαφορετικά, οι ασκήσεις πρέπει να παραδίδονται στους διδάσκοντες σε ηλεκτρονική μορφή μέσω του συνεργατικού συστήματος ηλεκτρονικής μάθησης `mooodle.softlab.ntua.gr`. Η προθεσμία παράδοσης θα τηρείται αυστηρά. Έχετε δικαίωμα να καθυστερήσετε το πολύ μία άσκηση.

Άσκηση 1 Haskell για ηχολήπτες

Προθεσμία παράδοσης: 1/11/2017

Γράψτε ένα πρόγραμμα σε Haskell που να λύνει το παρακάτω πρόβλημα. Υποβάλετε τη λύση σας στο σύστημα αυτόματης υποβολής και ελέγχου προγραμμάτων `grader.softlab.ntua.gr`.

Περιγραφή του προβλήματος. Δίνεται μια ακολουθία φυσικών αριθμών A_i (όπου $1 \leq i \leq N$). Σε μία συνεχόμενη υποπεριοχή των αριθμών, ας πούμε από L μέχρι R (όπου $1 \leq L \leq R \leq N$), μπορείτε να εφαρμόσετε έναν ενισχυτή. Ο ενισχυτής αυξάνει **το πολύ κατά M** όλους τους αριθμούς A_i που είναι στην περιοχή ($L \leq i \leq R$), αλλά μπορεί να χρησιμοποιηθεί **μόνο αν** όλοι αυτοί οι αριθμοί είναι ίσοι.

Γράψτε ένα πρόγραμμα που να υπολογίζει το ελάχιστο πλήθος ενισχυτών που πρέπει να εφαρμόσετε (τον έναν μετά τον άλλον) για να κάνετε ίσους όλους τους αριθμούς της ακολουθίας.

Προσέξτε γιατί το πλήθος των ενισχυτών μπορεί να είναι πολύ μεγάλο...

Είσοδος και έξοδος. Το πρόγραμμά σας θα διαβάζει τα δεδομένα από την τυπική είσοδο (`stdin`) και θα τυπώνει τα αποτελέσματα στην τυπική έξοδο (`stdout`).

Η πρώτη γραμμή της εισόδου περιέχει τους ακέραιους N και M , χωρισμένους με ένα κενό διάστημα. Η δεύτερη γραμμή περιέχει τους ακέραιους αριθμούς A_i , χωρισμένους με κενά διαστήματα.

Η μοναδική γραμμή της εξόδου πρέπει να περιέχει έναν μόνο ακέραιο: το ελάχιστο πλήθος ενισχυτών που κάνουν όλους τους αριθμούς της ακολουθίας ίσους.

Παράδειγμα εισόδου #1

3 2
1 3 2

Παράδειγμα εισόδου #2

4 1
1 2 4 2

Παράδειγμα εισόδου #3

5 2
3 1 4 1 1

Παράδειγμα εξόδου #1

2

Παράδειγμα εξόδου #2

5

Παράδειγμα εξόδου #3

4

Περιορισμοί.

- Για 50% των βαθμών, θα είναι $M = 1$
- Για 25% των βαθμών, θα είναι $1 \leq N \leq 10$
- Για 75% των βαθμών, θα είναι $1 \leq N \leq 10.000$
- Για 100% των βαθμών, θα είναι $1 \leq N \leq 200.000$, $1 \leq M \leq 10^9$ και $1 \leq A_i \leq 10^9$
- Όριο χρόνου εκτέλεσης: 1 sec.
- Όριο μνήμης: 64 MB.

Για να βαθμολογηθεί με άριστα, η λύση σας πρέπει να είναι αποδοτική. Προσέξτε ότι, σε μία γλώσσα αμιγούς συναρτησιακού προγραμματισμού όπως η Haskell, το αποδοτικό διάβασμα της εισόδου και η χρήση αποδοτικών δομών δεδομένων μπορεί να αποδειχθούν δύσκολο έργο.