

Άσκηση 1

Καταληκτική ημερομηνία και ώρα ηλεκτρονικής υποβολής: 29/4/2024, 23:59:59

Ήταν δίκαιο και έγινε... άσκηση (0.25+0.25 = 0.5 βαθμοί)

Δίνεται μια ακολουθία S από N ($2 \leq N \leq 1.000.000$) θετικών ακέραιους. Μια υπακολουθία S' αυτών των αριθμών η οποία ξεκινάει από τη θέση i και τελειώνει στη θέση j ($1 \leq i \leq j \leq N$) λέγεται «**δίκαιη**» για την S αν ελαχιστοποιεί την απόλυτη τιμή της διαφοράς του αθροίσματος των αριθμών που ανήκουν στην υπακολουθία S' μείον το άθροισμα των αριθμών της S που δεν ανήκουν στην υπακολουθία S' .

Η άσκηση σας ζητάει να γράψετε δύο προγράμματα (ένα σε C/C++ και ένα σε ML) τα οποία να διαβάζουν την είσοδο όπως φαίνεται παρακάτω και να τυπώνουν την απόλυτη τιμή της διαφοράς για κάποια δίκαιη υπακολουθία των αριθμών της εισόδου.

Τα στοιχεία εισόδου θα διαβάζονται από ένα αρχείο όπως φαίνεται στα παραδείγματα που ακολουθούν. Η πρώτη γραμμή του αρχείου περιέχει το πλήθος N των αριθμών της ακολουθίας. Η δεύτερη γραμμή του αρχείου περιέχει N μη αρνητικούς ακραίους χωρισμένους ανά δύο με ένα κενό διάστημα. Μπορείτε να υποθέσετε ότι το άθροισμα όλων των αριθμών της ακολουθίας δεν θα υπερβαίνει το 1.000.000.000.

Παρακάτω δίνονται τρία παραδείγματα σε C/C++ και σε ML. Στο πρώτο παράδειγμα, η δίκαιη υπακολουθία είναι η **2 3**. Στο δεύτερο παράδειγμα, τόσο η **1 3 4** η **3 4** όσο και η **2 5** είναι δίκαιες. Στο τρίτο παράδειγμα, το **42** είναι η μία από τις δύο δίκαιες υπακολουθίες και οι υπόλοιποι αριθμοί αποτελούν την άλλη.

Σε C/C++, MLton, ή σε OCaml

```
$ ./fairseq f1.txt  
0
```

```
$ ./fairseq f2.txt  
1
```

```
$ ./fairseq f3.txt  
17
```

Σε SML/NJ

```
- fairseq "f1.txt";  
0
```

```
val it = () : unit
```

```
- fairseq "f2.txt";  
1
```

```
val it = () : unit
```

```
- fairseq "f3.txt";  
17
```

```
val it = () : unit
```

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

```
$ cat f1.txt
```

```
4
```

```
1 2 3 4
```

```
$ cat f2.txt
```

```
5
```

```
1 3 4 2 5
```

```
$ cat f3.txt
```

```
5
```

```
1 17 5 2 42
```

Το Δένδρο του Κόσμου (0.25+0.25 = 0.5 βαθμοί)

Στην καρδιά ενός μυθικού δάσους υπάρχει ένα θρυλικό δέντρο, γνωστό ως Δέντρο του Κόσμου. Το Δέντρο του Κόσμου είναι ένα υπέροχο δυαδικό δέντρο με N κόμβους χαραγμένους με μοναδικούς αριθμούς από το 1 ως το N . Αυτοί οι αριθμοί κρατούν τα μυστικά για την ισορροπία του σύμπαντος.

Συγκεκριμένα, η ακολουθία των κόμβων που προκύπτει από την ενδοδιατεταγμένη (in-order) διάσχιση του δέντρου καθορίζει την αρμονία μεταξύ των ουρανών και της γης. Σε κάθε κόμβο αυτού του δέντρου μπορούμε να εφαρμόσουμε μια πράξη αντιμετάθεσης (swap operation), που αντιμεταθέτει τα παιδιά του: το αριστερό παιδί (αν φυσικά υπάρχει) γίνεται δεξιό και το δεξιό παιδί (αν πάλι υπάρχει) γίνεται αριστερό. Κατ' αυτόν τον τρόπο αλλάζει η ακολουθία που προκύπτει από την ενδοδιατεταγμένη διάσχιση του δέντρου και, κατ' επέκταση, αλλάζουν οι ισορροπίες του σύμπαντος.

Η ιέρεια του μυθικού δάσους, απογοητευμένη από τη σημερινή κατάσταση των ισορροπιών του σύμπαντος, αποφασίζει να αναθέσει στον Βαγγέλη, έναν φοιτητή της ΣΗΜΜΥ γνωστό για τις γνώσεις του στα δυαδικά δέντρα, να εφαρμόσει πράξεις αντιμετάθεσης σε όσους κόμβους του δέντρου χρειάζεται, ούτως ώστε η ακολουθία των κόμβων που θα προκύπτει από την ενδοδιατεταγμένη διάσχιση να είναι λεξικογραφικά η ελάχιστη δυνατή. Ο στόχος σας είναι να βοηθήσετε τον Βαγγέλη να επαναφέρει την κοσμική τάξη.

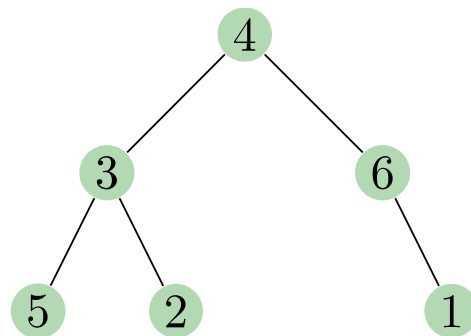
Η άσκηση σας ζητάει να γράψετε δύο προγράμματα (ένα σε C/C++ και ένα σε ML) τα οποία να διαβάζουν το αρχικό δέντρο και να τυπώνει τη λεξικογραφικά ελάχιστη δυνατή ακολουθία των κόμβων που μπορεί να προκύψει από την ενδοδιατεταγμένη διάσχιση του δέντρου, μετά από την εφαρμογή οσωνδήποτε κατάλληλων πράξεων αντιμετάθεσης στους κόμβους του δέντρου.

Τα στοιχεία εισόδου θα διαβάζονται από ένα αρχείο όπως φαίνεται στο παράδειγμα που ακολουθεί. Η πρώτη γραμμή του αρχείου θα περιέχει έναν φυσικό αριθμό, το πλήθος N των κόμβων του δέντρου ($1 \leq N \leq 1.000.000$). Η δεύτερη γραμμή θα προκύπτει από την παρακάτω γραμματική σε μορφή EBNF και θα περιγράφει τη δομή του δέντρου. Το σύμβολο "0" παριστάνει ένα κενό δέντρο, ενώ το σύμβολο <number> παριστάνει έναν μοναδικό αριθμό με τιμή από 1 μέχρι και N .

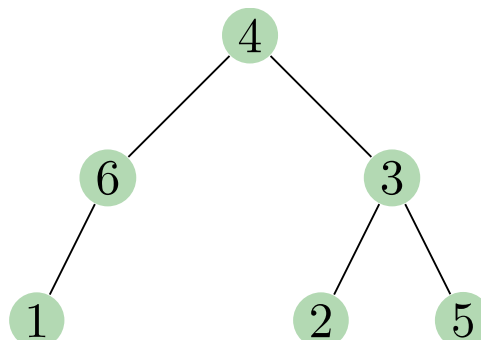
```
<tree> ::= 0
          | <number> <tree> <tree>
```

Δείτε το παρακάτω παράδειγμα εισόδου και το δέντρο που της αντιστοιχεί.

tree1.txt												
6												
4	3	5	0	0	2	0	0	6	0	1	0	0



Ο Βαγγέλης αποφασίζει να εφαρμόσει την πράξη αντιμετάθεσης στους κόμβους 4, 6 και 3, οπότε το δέντρο γίνεται όπως φαίνεται στο παρακάτω σχήμα. Η ακολουθία που προκύπτει από την ενδοδιατεταγμένη διάταξή του (αγνοώντας αυτή τη φορά τα κενά δέντρα) είναι: 1 6 4 2 3 5. Αυτή είναι λεξικογραφικά η ελάχιστη δυνατή που μπορεί να προκύψει.



Το πρόγραμμα θα πρέπει να τυπώνει τη λεξικογραφικά ελάχιστη ακολουθία που μπορεί να προκύψει από την ενδοδιατεταγμένη διάσχιση του δέντρου, μετά από την εφαρμογή οσωνδήποτε κατάλληλων πράξεων αντιμετάθεσης στους κόμβους του δέντρου. Ακολουθεί παράδειγμα εκτέλεσης σε C/C++ και σε ML.

Σε C/C++, MLton, ή σε OCaml

```
$ ./arrange tree1.txt  
1 6 4 2 3 5
```

Σε SML/NJ

```
- arrange "tree1.txt";  
1 6 4 2 3 5  
val it = () : unit
```

Περαιτέρω οδηγίες για τις ασκήσεις

- Μπορείτε να δουλέψετε σε ομάδες το πολύ δύο ατόμων, τόσο σε αυτή όσο και στις επόμενες σειρές ασκήσεων. Όμως, έχετε υπ' όψη σας ότι, αν δεν περάσετε το μάθημα φέτος, οι βαθμοί των προγραμματιστικών ασκήσεων κρατούνται μόνο για όσους δεν τις έκαναν σε ομάδα αλλά τις έκαναν μόνοι τους. (Για όλους τους υπόλοιπους, οι βαθμοί των ασκήσεων κρατούνται.)
- 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. κώδικα ταξινόμησης, 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 (link) σε αυτόν.
- Τα προγράμματα σε C/C++ πρέπει να είναι σε ένα αρχείο και να μπορούν να μεταγλωττιστούν χωρίς warnings με gcc/g++ (version ≥ 10.2.1) με εντολές της μορφής, π.χ.

```
gcc -std=c99 -Wall -Werror -O3 -lm -o fairseq fairseq.c  
g++ -std=c++11 -Wall -Werror -O3 -lm -o fairseq fairseq.cpp
```

- Τα προγράμματα σε ML πρέπει επίσης να είναι σε ένα αρχείο και να δουλεύουν σε SML/NJ ≥ v110.79 ή σε MLton ≥ 20210117 ή σε Objective Caml version ≥ 4.11.1. Το σύστημα ηλεκτρονικής υποβολής σας επιτρέπει να επιλέξετε μεταξύ αυτών των διαλέκτων της ML.
- Η υποβολή των προγραμμάτων θα γίνει ηλεκτρονικά μέσω του helios και για να μπορέσετε να τα υποβάλλετε και να βαθμολογηθείτε για αυτά, τα μέλη της ομάδας σας (και οι δύο) θα πρέπει να έχετε εγγραφεί στο μάθημα στο helios. Θα υπάρξει σχετική ανακοίνωση για την ακριβή διαδικασία υποβολής όταν ανοίξει το σύστημα. Τα προγράμματά σας πρέπει να διαβάζουν την είσοδο όπως αναφέρεται και δεν πρέπει να έχουν κάποιου άλλους είδους έξοδο διότι δεν θα γίνουν δεκτά από το σύστημα στο οποίο θα υποβληθούν.