



Γλώσσες Προγραμματισμού II

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

Άσκηση 1 Haskell για ενεργειακούς

Προθεσμία παράδοσης: 3/11/2010

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. Πλησιάζουν τα Χριστούγεννα και ο μικρός Γουσταύος Κίρχοφ (εις μάτην της φυσιολογικής ροής της ιστορίας) ετοιμάζεται να στολίσει τον κήπο του με πολύχρωμους ηλεκτρικούς λαμπτήρες. Ο κήπος του αποτελείται από N θάμνους (αριθμημένους από 1 έως N) και σε κάθε έναν από αυτούς, ανάλογα με το μέγεθος του φυλλώματος, ο Γουσταύος έχει τοποθετήσει b_i όμοιους λαμπτήρες ($1 \leq i \leq N$). Επειδή τα οικονομικά του δεν πάνε πολύ καλά μετά την έλευση του ΔΝΤ, ο Γουσταύος αποφάσισε να συνδέσει τους θάμνους μεταξύ τους με όσο το δυνατόν λιγότερα καλώδια ρεύματος. Ο κολλητός του φίλος Λεονάρδος (πάλι εις μάτην της φυσιολογικής ροής της ιστορίας) του εξήγησε ότι αρκούν $N - 1$ καλώδια, τα οποία ουσιαστικά ορίζουν ένα δέντρο — το δέντρο των θάμνων! Ο Γουσταύος που είναι καλός στα ρεύματα αλλά σκράπας στη γραφοθεωρία και στην κηπουρική, δεν ήταν σίγουρος ότι κατάλαβε αλλά έσπευσε να τοποθετήσει τα καλώδια.

Εκεί όμως που είχε συνδέσει τον κοντινότερο θάμνο στην πηγή του ρεύματος και ετοιμαζόταν να σηκώσει το διακόπτη, ο Γουσταύος συνέλαβε τον περίφημο πρώτο κανόνα: “σε κάθε κόμβο ενός ηλεκτρικού κυκλώματος, το άθροισμα των εισερχόμενων ρευμάτων ισούται με το άθροισμα των εξερχόμενων ρευμάτων.” Σκέφτηκε λίγο καλύτερα και ευχαρίστησε την τύχη του που δε σήκωσε το διακόπτη, διαφορετικά κάποια από τα καλώδιά του θα έλιωναν λόγω του υπερβολικού ρεύματος που θα περνούσε από αυτά. Βοηθήστε τον να βρει σε ποιο θάμνο πρέπει να συνδέσει την πηγή ώστε το πιο “φορτωμένο” καλώδιο να έχει το λιγότερο δυνατό ρεύμα.

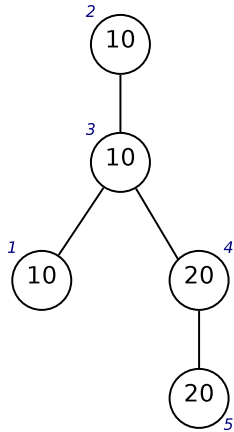
Είσοδος και έξοδος. Το πρόγραμμά σας θα διαβάζει τα δεδομένα από την τυπική είσοδο (stdin) και θα τυπώνει τα αποτελέσματα στην τυπική έξοδο (stdout).

Η πρώτη γραμμή της εισόδου θα περιέχει το φυσικό αριθμό N . Οι επόμενες N γραμμές αντιστοιχούν στους θάμνους του κήπου, κατά αύξουσα σειρά — η δεύτερη γραμμή στο θάμνο με αριθμό 1, η τρίτη σε αυτόν με αριθμό 2, κ.ο.κ. Σε κάθε μία από αυτές θα υπάρχουν δύο φυσικοί αριθμοί b_i και p_i (όπου $1 \leq i \leq N$). Το b_i είναι το πλήθος των λαμπτήρων στο θάμνο i . Το p_i είναι ο αριθμός του θάμνου που έχει τη θέση του “προγόνου” του i στο δέντρο των θάμνων. Αν ο θάμνος i είναι η ρίζα του δέντρου των θάμνων, θα είναι $p_i = 0$.

Το πρόγραμμά σας πρέπει να τυπώνει τον αριθμό του θάμνου όπου πρέπει να συνδεθεί η πηγή του ρεύματος. Αν υπάρχουν περισσότερες βέλτιστες λύσεις, πρέπει να τυπώνει το θάμνο με το μικρότερο αριθμό.

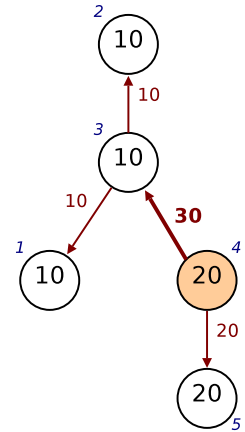
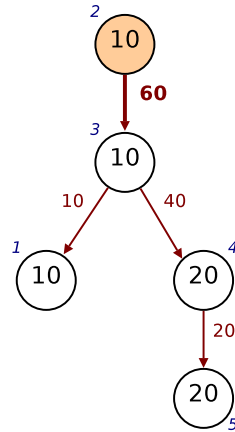
Παράδειγμα εισόδου.

5
10 3
10 0
10 2
20 3
20 4



Παράδειγμα εξόδου.

4



Εξήγηση. Το δέντρο των θάμνων φαίνεται στο αριστερό σχήμα. Αν η πηγή ρεύματος συνδεθεί στο θάμνο 2, τότε τα ρεύματα που διατρέχουν τα καλώδια φαίνονται στο μεσαίο σχήμα. Το μέγιστο φορτίο είναι 60, στο καλώδιο μεταξύ των θάμνων 2 και 3. Αν όμως η πηγή ρεύματος συνδεθεί στο θάμνο 4, τότε προκύπτουν τα ρεύματα που φαίνονται στο δεξιό σχήμα. Το μέγιστο φορτίο είναι 30, στο καλώδιο μεταξύ των θάμνων 4 και 3. Αυτή είναι και η βέλτιστη λύση.

Περιορισμοί. Για να βαθμολογηθεί με άριστα, η λύση σας πρέπει να είναι αποδοτική. Περιορισμοί στο μέγεθος της εισόδου, στο χρόνο εκτέλεσης και στη μνήμη αναγράφονται στο σύστημα αυτόματης υποβολής και ελέγχου. Σωστές λύσεις με πολυπλοκότητα $O(N^2)$ θα παίρνουν περίπου τις μισές μονάδες. Προσέξτε ότι, σε μία γλώσσα αμιγούς συναρτησιακού προγραμματισμού όπως η Haskell, το αποδοτικό διάβασμα της εισόδου και η κατασκευή της επιθυμητής αναπαράστασης του δέντρου των θάμνων μπορεί να αποδειχθεί δύσκολο έργο.