



Προγραμματιστικές Τεχνικές

Άσκηση 10 Δυαδικά Δένδρα

Προθεσμία υποβολής στον grader: 30/5/2022

Υλοποιήστε την κλάση `lexicon` για την αναπαράσταση λεξιλογίων κειμένων με τη μορφή δυαδικών δένδρων αναζήτησης. Η κλάση θα πρέπει να υποστηρίζει τις παρακάτω λειτουργίες, τις οποίες πρέπει να υλοποιήσετε:

```
1 class lexicon {  
2 public:  
3     lexicon();  
4     ~lexicon();  
5  
6     void insert(const string &s);  
7     int lookup(const string &s) const;  
8     int depth(const string &s) const;  
9     void replace(const string &s1, const string &s2);  
10  
11     friend ostream & operator << (ostream &out, const lexicon &l);  
12 };
```

Θεωρήστε δεδομένο ότι οι λέξεις που θα αποθηκεύονται σε ένα λεξικό δεν θα είναι κενές και θα περιέχουν μόνο μικρά λατινικά γράμματα. Κάθε λέξη θα αντιστοιχεί σε έναν κόμβο στο δένδρο. Ο κόμβος θα περιέχει τη λέξη και τη συχνότητα εμφάνισής της στο λεξικό (δηλαδή έναν μετρητή που θα θυμάται πόσες φορές έχει εισαχθεί αυτή η λέξη). Το αριστερό παιδί του κόμβου θα περιέχει αλφαβητικά (λεξικογραφικά) μικρότερες λέξεις, ενώ το δεξιό παιδί αλφαβητικά (λεξικογραφικά) μεγαλύτερες. Καμία λέξη δεν θα πρέπει να εμφανίζεται σε περισσότερους από έναν κόμβους στο δένδρο. Επίσης, το δένδρο θα πρέπει να είναι ένα απλό BST: δεν απαιτείται και δεν πρέπει να είναι ισοζυγισμένο (π.χ., AVL).

Η μέθοδος `insert(s)` θα εισάγει τη λέξη `s` στο δένδρο.

Η μέθοδος `lookup(s)` θα αναζητά τη λέξη `s` στο δένδρο και θα επιστρέφει τη συχνότητα εμφάνισής. Το αποτέλεσμα θα είναι 0 (μηδέν) αν η λέξη δεν υπάρχει στο δένδρο.

Η μέθοδος `depth(s)` θα αναζητά τη λέξη `s` στο δένδρο, όπως και η `lookup`. Αν δεν υπάρχει, θα επιστρέφει `-1`. Αν όμως υπάρχει, θα επιστρέφει το βάθος στο οποίο βρίσκεται ο κόμβος που την περιέχει στο δένδρο, δηλαδή το μήκος του μονοπατιού από τη ρίζα έως αυτόν τον κόμβο. Θεωρούμε ότι η ρίζα του δένδρου βρίσκεται σε βάθος 0 (μηδέν).

Η μέθοδος `replace(s1, s2)` Θα αντικαθιστά όλες τις εμφανίσεις της λέξης `s1` με ισάριθμες εμφανίσεις της λέξης `s2`. Αν η `s1` δεν υπάρχει στο δένδρο, τότε δε θα γίνεται τίποτα. Αν υπάρχει και το πλήθος εμφανίσεών της είναι $k > 0$, τότε η `s1` θα διαγράφεται και θα αναζητάται η λέξη `s2`. Αν αυτή δεν υπάρχει, θα εισάγεται με συχνότητα εμφάνισης k . Αν όμως υπάρχει, τότε η συχνότητα εμφάνισής της θα ενημερώνεται κατάλληλα (θα αυξάνει κατά k).

Κατά τη διαγραφή μίας λέξης από το λεξικό (που γίνεται έμμεσα με τη μέθοδο `replace`), αν διαγράφεται κόμβος που έχει δύο μη κενά παιδιά, τότε ο κόμβος που διαγράφεται αντικαθίσταται από αυτόν

που περιέχει την αμέσως μικρότερη λέξη. Αν διαγράφεται κόμβος που έχει ένα μη κενό παιδί, τότε αντικαθίσταται από το παιδί του.

Η εκτύπωση των λεξικών πρέπει να γίνεται σε αλφαβητική σειρά, με τις λέξεις να ακολουθούνται από τη συχνότητα εμφάνισής τους.

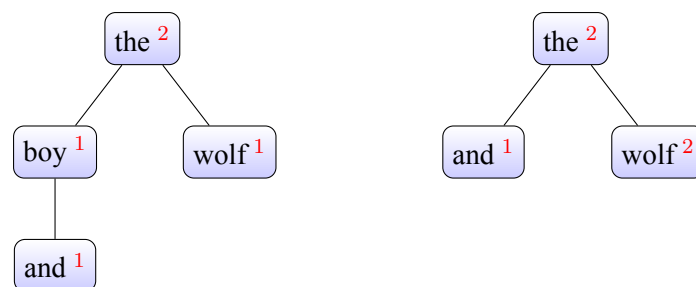
Μπορείτε να δοκιμάσετε την υλοποίησή σας με προγράμματα όπως το εξής:

```
1 int main() {
2     lexicon l;
3     l.insert("the");
4     l.insert("boy");
5     l.insert("and");
6     l.insert("the");
7     l.insert("wolf");
8     cout << "The word 'the' is found " << l.lookup("the") << " time(s)" << endl;
9     cout << "The word 'and' is found at depth " << l.depth("and") << endl;
10    cout << l;
11    l.replace("boy", "wolf");
12    cout << "After replacement:\n";
13    cout << l;
14    cout << "Now the word 'and' is found at depth " << l.depth("and") << endl;
15 }
```

Η εκτέλεσή του θα πρέπει να εμφανίζει:

```
1 The word 'the' is found 2 time(s)
2 The word 'and' is found at depth 2
3 and 1
4 boy 1
5 the 2
6 wolf 1
7 After replacement:
8 and 1
9 the 2
10 wolf 2
11 Now the word 'and' is found at depth 1
```

Η μορφή του δένδρου πριν (αριστερά) και μετά (δεξιά) την κλήση της replace δίνεται παρακάτω.



Στο αρχείο που θα ανεβάσετε στον grader θα πρέπει να συμπεριλάβετε (μόνο) τη δήλωση της κλάσης lexicon και τις υλοποιήσεις των μεθόδων της.

Άσκηση 11 Ενώστε τα χωριά

Προθεσμία υποβολής στον grader: 30/5/2022

Μετά το 144ο 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 ανεξάρτητες 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.

Το κράτος, με σύνθημα “έρχεται η επανασύνδεση”, σχεδιάζει την ανασυγκρότηση του οδικού δικτύου. Όμως, λόγω περιορισμένων πόρων 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 ένα πρόγραμμα που να βρίσκει το ελάχιστο δυνατό πλήθος ανεξάρτητων ομάδων που μπορεί να μείνει μετά την κατασκευή των νέων δρόμων. Θα πρέπει να διαβάσει τους υπάρχοντες δρόμους από το standard input και να εκτυπώνει στο standard output μία γραμμή που να περιέχει έναν μόνο ακέραιο αριθμό: το ελάχιστο δυνατό πλήθος ομάδων.

Η είσοδος θα περιέχει τα εξής. Η πρώτη γραμμή 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 ήδη, πριν την κατασκευή νέων δρόμων, ένας βατός δρόμος που συνδέει τα χωριά A και B . Θεωρήστε ότι τα χωριά είναι αριθμημένα από 1 μέχρι και N .

Παραδείγματα:

Είσοδος:

7 2 2	4 2 3	4 3 0
1 2	1 2	3 2
6 5	4 3	1 4
		1 3

Έξοδος:

3	1	1
---	---	---

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 ($K = 0$).

Σημείωση: Για την άσκηση αυτή, θα σας είναι χρήσιμη η δομή union-find. Ξεκινήστε με N ομάδες αποτελούμενες από 1 χωριό κάθε μία και ενώστε, προσθέτοντας έναν-έναν τους υπάρχοντες δρόμους. Μετρήστε πόσες ομάδες υπάρχουν αρχικά. Μετά, σκεφτείτε σαν μηχανικοί! ☺