

Αλγόριθμοι και Πολυπλοκότητα
Προγραμματιστική Άσκηση
Παράδοση: Δευτέρα 13 Ιουλίου 2015

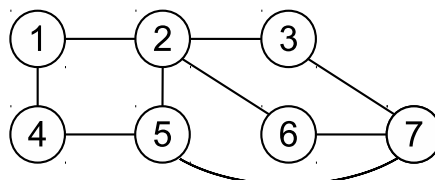
Γενικά.

1. Η άσκηση είναι προαιρετική και μπορεί να γίνει ατομικά ή σε ζεύγη.
2. Το βάρος της άσκησης στον τελικό βαθμό είναι $(40/3)\%$.
3. Για την υλοποίηση των προγραμμάτων μπορείτε να χρησιμοποιήσετε τις γλώσσες C/C++ και Java.
4. Επιτρέπεται να χρησιμοποιήσετε ψευδοκώδικα από τα προτεινόμενα συγγράμματα και τις διαφάνειες.
5. Ο κώδικας θα πρέπει να έχει επαρκή σχόλια. Ειδικά θα πρέπει να δώσετε με τη μορφή σχολίων στον κώδικα σύντομες περιγραφές των αλγορίθμων σας και να αναφέρετε τους αντίστοιχους χρόνους εκτέλεσης.
6. Στην αρχή κάθε αρχείου κώδικα θα πρέπει να εισάγετε σχόλιο με τα στοιχεία σας (ονοματεπώνυμο, ΑΜ) και με τα στοιχεία του υπολογιστικού περιβάλλοντος όπου εκτελέστηκε το πρόγραμμα (τύπος και ταχύτητα επεξεργαστή, μεταγλωττιστής, λειτουργικό σύστημα).
7. Η παράδοση της άσκησης γίνεται ηλεκτρονικά μέσω της πλατφόρμας eclass.
8. Παραδοτέα είναι αποκλειστικά και μόνο τα αρχεία του πηγαίου κώδικα των δύο προγραμμάτων συμπιεσμένα σε ένα αρχείο zip.

Πρόβλημα 1.

Υλοποιήστε τον αλγόριθμο αναζήτησης κατά πλάτος (BFS) για ένα μη κατευθυνόμενο συνδεδεμένο γράφημα G . Τροποποιήστε τον αλγόριθμο ώστε να υπολογίζει επιπρόσθετα (α) το πλήθος των κόμβων σε κάθε επίπεδο του δέντρου BFS, (β) το αν υπάρχει ή όχι οριζόντια ακμή, δηλαδή αν υπάρχει μία ακμή που να συνδέει δύο κόμβους σε κοινό επίπεδο του δέντρου BFS, (γ) το βάθος του δέντρου BFS, και (δ) ένα συντομότερο μονοπάτι από τη ρίζα σε έναν κόμβο μέγιστου βάθους.

Το γράφημα εισόδου πρέπει να διαβάζεται από ένα αρχείο με όνομα `graph.in` που έχει την ακόλουθη μορφή. Στην πρώτη γραμμή του αρχείου δίνεται το πλήθος των κόμβων n . Θεωρήστε ότι οι κόμβοι είναι αριθμημένοι από 1 έως n . Στη συνέχεια ακολουθούν n γραμμές. Στην i -οστή γραμμή περιέχονται στη σειρά οι γειτονικοί κόμβοι του κόμβου i . Συγκεκριμένα ο πρώτος ακέραιος d αντιστοιχεί στο βαθμό του κόμβου i και οι επόμενοι d ακέραιοι στους γειτονικούς του κόμβους.



Για παράδειγμα, ένα αρχείο εισόδου που αντιστοιχεί στο γράφημα του σχήματος είναι το εξής:

```
7
2 2 4
4 1 5 3 6
```

```

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

```

Το πρόγραμμα πρέπει να διαβάζει και να αποθηκεύει το G στη μνήμη σε χώρο $O(m)$ όπου m το πλήθος των ακμών, να εκτελεί τον αλγόριθμο BFS με αφετηρία τον κόμβο 1, να υπολογίζει τα ζητούμενα των (α) έως (δ), να εκτυπώνει τα αποτελέσματα στην οθόνη, και κατόπιν να τερματίζει. Δίνουμε μία ενδεικτική έξοδο την οποία το πρόγραμμα θα μπορούσε να παράγει με είσοδο το παραπάνω γράφημα:

```

Number of nodes at each level (from 1 to 3):
2, 3, 1
No horizontal edge.
Tree depth:
3
Shortest path to node of maximum depth:
1, 4, 5, 7

```

Πρόβλημα 2.

Έστω ότι επιθυμούμε να μετρήσουμε με κάποιο τρόπο την ομοιότητα δύο συμβολοσειρών. Γενικά μία στοίχιση δύο συμβολοσειρών είναι η αναγραφή της μίας συμβολοσειράς πάνω από την άλλη χρησιμοποιώντας ενδεχομένως και παύλες έτσι ώστε κάθε χαρακτήρας μίας συμβολοσειράς να αντιστοιχείται στην ίδια στήλη με έναν χαρακτήρα της άλλης συμβολοσειράς ή μία παύλα. Δείτε για παράδειγμα μία πιθανή στοίχιση για τις συμβολοσειρές ABDECFAB και ABCDEFC.

A	B	—	D	E	C	F	A	B
A	B	C	D	E	F	C	—	—

Το κόστος μίας στοίχισης είναι το άθροισμα του κόστους των στηλών το οποίο υπολογίζεται ως εξής. Αν σε μία στήλη υπάρχουν δύο ίδιοι χαρακτήρες τότε το κόστος στήλης είναι 0 ενώ αν υπάρχουν δύο διαφορετικοί χαρακτήρες ή παύλα το κόστος είναι 1. Με βάση τα παραπάνω εύκολα προκύπτει ότι η στοίχιση του προηγούμενου παραδείγματος έχει κόστος πέντε.

Να σχεδιάσετε και να υλοποιήσετε έναν αλγόριθμο δυναμικού προγραμματισμού για την εύρεση του ελάχιστου κόστους στοίχισης δύο οποιωνδήποτε συμβολοσειρών. (Όσο μικρότερη η τιμή κόστους τόσο περισσότερο όμοιες θεωρούνται οι δύο συμβολοσειρές.) Σημειώνεται ότι το κόστος αυτό είναι γνωστό και ως η διορθωτική απόσταση (edit distance) μεταξύ των δύο συμβολοσειρών.

Το πρόγραμμά σας θα πρέπει να διαβάζει από την τυπική είσοδο σε δύο διαφορετικές γραμμές δύο συμβολοσειρές με μήκος έως 100 χαρακτήρες και να τυπώνει στην τυπική έξοδο έναν ακέραιο ίσο με το ελάχιστο κόστος στοίχισης των δύο συμβολοσειρών.