

HY380 – Αλγόριθμοι και πολυπλοκότητα

Hard Problems

Ημερομηνία Παράδοσης: 08/05/2017

την ώρα του μαθήματος ή email:

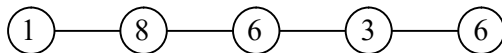
Giannis Agathangelos <jagathan@csd.uoc.gr>

Γενικές Οδηγίες

- α) Επιτρέπεται η αναζήτηση στο Internet και στην βιβλιοθήκη αλλά αν χρησιμοποιήσετε κάτι θα πρέπει να το αναφέρετε στην λύση του αντίστοιχου προβλήματος.
- β) Προσπαθήστε να λύσετε όσον το δυνατόν περισσότερα από τα προβλήματα.
- γ) Η προσπάθεια βαθμολογείται με την προϋπόθεση ότι θα φαίνεται καθαρά ποια είναι η στρατηγική που ακολουθήσατε για την λύση ενός προβλήματος.

Πρόβλημα 1

Έστω $G = (V, E)$ ένας μη κατευθυνόμενος γράφος με n κόμβους. Ένα υποσύνολο των κόμβων ονομάζεται ανεξάρτητο (independent set) σύνολο αν δεν υπάρχουν δύο κόμβοι του υποσυνόλου που να συνδέονται με ακμή. Η εύρεση ενός ανεξάρτητου συνόλου με μέγιστο αριθμό κόμβων είναι δύσκολο πρόβλημα στην γενική του περίπτωση. Στο πρόβλημα αυτό θα δούμε ότι η λύση μπορεί να βρεθεί με δραστικό αλγόριθμο αν ο γράφος είναι αρκετά απλός. Ένας γράφος $G = (V, E)$ ονομάζεται μονοπάτι εάν οι κόμβοι του μπορούν να γραφούν ως v_1, v_2, \dots, v_n , με μια ακμή μεταξύ των κόμβων v_i και v_j εάν και μόνο εάν οι αριθμοί i και j διαφέρουν κατά ένα. Σε κάθε κόμβο v_i αντιστοιχίζουμε ένα βάρος που είναι ένας θετικός ακέραιος. π.χ. Θεωρήστε το μονοπάτι του επόμενου σχήματος. Τα βάρη είναι οι αριθμοί μέσα σε κάθε κόμβο.



Ο στόχος είναι να λυθεί το επόμενο αλγοριθμικό πρόβλημα:

Να βρεθεί ένα ανεξάρτητο σύνολο μέγιστου βάρους σε ένα μονοπάτι G .

- α) Δείξτε με ένα παράδειγμα ότι ο επόμενος αλγόριθμος δεν βρίσκει πάντα ένα ανεξάρτητο σύνολο μέγιστου βάρους.

The “heaviest-first” greedy algorithm:

1. Start with S equal to empty set
2. **while** some node remains in G

3. Pick a node v_i of maximum weight
4. Add v_i to S
5. Delete v_i and its neighbors from G
6. **end while**
7. **return** S

β) Δείξτε με ένα παράδειγμα ότι και ο επόμενος αλγόριθμος δεν βρίσκει πάντα ένα ανεξάρτητο σύνολο μέγιστου βάρους.

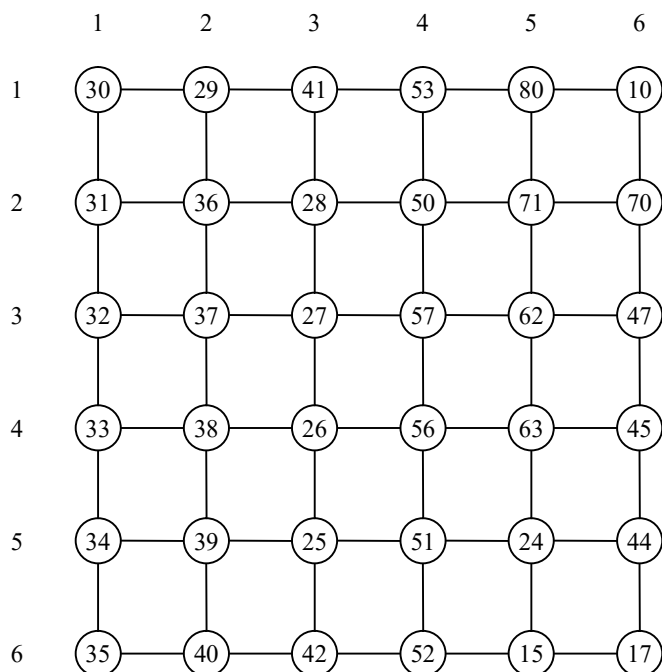
1. Let S_1 be the set of all v_i where i is an odd number.
2. Let S_2 be the set of all v_i where i is an even number.
/* Note that S_1 and S_2 are both independent sets. */
3. Determine which of S_1 or S_2 has greater total weight, and return this one.

γ) Δώστε έναν αλγόριθμο ο οποίος δέχεται ως είσοδο ένα μονοπάτι G με n κόμβους με βάρη και επιστρέφει ως έξοδο ένα ανεξάρτητο σύνολο μέγιστου βάρους. Ο χρόνος που απαιτεί ο αλγόριθμος θα πρέπει να είναι ένα πολυώνυμο ως προς το n και να μην εξαρτάται από τα βάρη.

Πρόβλημα 2

Ένα $n \times n$ πλέγμα είναι ένας γράφος G του οποίου οι κόμβοι είναι διατεταγμένα ζεύγη φυσικών αριθμών (i, j) με $1 \leq i \leq n$ και $1 \leq j \leq n$; Δύο κόμβοι (i_1, j_1) και (i_2, j_2) είναι γείτονες εάν και μόνο εάν $|i_1 - i_2| + |j_1 - j_2| = 1$.

Υποθέτουμε ότι σε κάθε κόμβο έχει αντιστοιχηθεί ένας μοναδικός φυσικός αριθμός x_v . Ένας κόμβος v είναι τοπικό ελάχιστο εάν $x_v < x_u$ για όλους τους γείτονες u του v . Για παράδειγμα στον γράφο του επόμενου σχήματος υπάρχουν τέσσερα τοπικά ελάχιστα. Οι κόμβοι $(1, 2)$, $(1, 6)$, $(5, 3)$, $(6, 5)$ με ετικέτες τους φυσικούς αριθμούς 29, 10, 25, 15 αντίστοιχα.



Δοθέντος ενός τέτοιου γράφου G και μιας αντιστοίχισης φυσικών αριθμών (ετικετών) στους κόμβους του, το πρόβλημα είναι να βρούμε ένα τοπικό ελάχιστο του G .

α) Θεωρείστε τον επόμενο αλγόριθμο ο οποίος αρχίζει από τον κόμβο $(1,1)$ και ψάχνει για τον μικρότερο γείτονα μέχρι να βρεθεί ένα τοπικό ελάχιστο του G .

1. $v \leftarrow (1,1)$
2. **while** there is a neighbor u of v such that $x_u < x_v$
3. Let u be the neighbor of v with smallest x_u
4. $v \leftarrow u$
5. **end while**
6. **return** v

Να δείξετε ότι ο παραπάνω αλγόριθμος πάντα επιστρέφει ένα τοπικό ελάχιστο του G . Πόσο χρόνο χρειάζεται ο αλγόριθμος στη χειρίστη περίπτωση. Να δώσετε την απάντησή σας με τον συμβολισμό Θ .

β) Να δώσετε έναν αλγόριθμο ο οποίος λύνει το πρόβλημα σε χρόνο $O(n)$. Να δείξετε ότι ο αλγόριθμός σας πάντα επιστρέφει ένα τοπικό ελάχιστο. Υποθέστε ότι το n είναι δύναμη του 2 ($n = 2^k$)

Πρόβλημα 3

Δίνονται τρία διανύσματα A , B και C διάστασης n με στοιχεία πραγματικούς αριθμούς. Γράψτε έναν αλγόριθμο ο οποίος βρίσκει αν υπάρχουν τρεις αριθμοί, ένας από κάθε έναν από τους πίνακες A , B και C , οι οποίοι έχουν άθροισμα 0.

Μπορείτε να βρείτε έναν $O(n^3)$ αλγόριθμο;

Μπορείτε να βρείτε έναν $O(n^2 \log n)$ αλγόριθμο;

Μπορείτε να βρείτε έναν $O(n^2)$ αλγόριθμο;

Μπορείτε να βρείτε έναν αλγόριθμο ταχύτερο από $O(n^2)$;

Πρόβλημα 4

Γράψτε μια έκδοση του Fast Fourier Transform (fft) για την περίπτωση που το n είναι δύναμη του 3, χωρίζοντας το διάνυσμα εισόδου σε τρία υποδιανύσματα., λύνοντας το πρόβλημα αναδρομικά σε αυτά και συνδυάζοντας τις λύσεις των υποπροβλημάτων. Γράψτε την αναδρομική σχέση για τον χρόνο εκτέλεσης του προγράμματος και λύστε την αναδρομική σχέση με το κεντρικό θεώρημα.

Πρόβλημα 5

Δίνεται μη κατευθυνόμενο γράφος $G = (V, E)$, και δύο κόμβοι v και u του G . Δώστε έναν αλγόριθμο ο οποίος να υπολογίζει τον αριθμό των συντομότερων v - u μονοπατιών στον G . (Ο αλγόριθμος δεν χρειάζεται να τυπώνει όλα τα μονοπάτια, αλλά μόνο το πόσα υπάρχουν). Ο αλγόριθμος θα πρέπει να τρέχει σε χρόνο $O(n + m)$ για έναν γράφο με n κορυφές και m ακμές.

Πρόβλημα 6

Θεωρούμε έναν μακρύ ευθύγραμμο δρόμο, κατά μήκος του οποίου υπάρχουν κάποια σπίτια. Μια εταιρεία κινητής τηλεφωνίας θέλει να τοποθετήσει κεραίες κατά μήκος του δρόμου για την εξυπηρέτηση των κατοίκων των σπιτιών. Η εμβέλεια κάθε κεραίας είναι τέσσερα χιλιόμετρα. Δώστε έναν αλγόριθμο ο οποίος να τοποθετεί τον μικρότερο αριθμό κεραιών.

Πρόβλημα 7

Σε κάποιες εφαρμογές θέλουμε να βρούμε ελαφριά συνδετικά δένδρα με συγκεκριμένες ιδιότητες. Το επόμενο πρόβλημα είναι ένα παράδειγμα.

Είσοδος: Μη κατευθυνόμενος γράφος $G = (V, E)$, βάρη ακμών w_e , και ένα υποσύνολο U των κόμβων ($U \subset V$).

Έξοδος: Το ελαφρύτερο συνδετικό δένδρο στο οποίο οι κόμβοι του συνόλου U είναι φύλλα. (Το δένδρο μπορεί να έχει φύλλα και από τους κόμβους του συνόλου $V - U$)

Να γράψετε έναν αλγόριθμο για το παραπάνω πρόβλημα ο οποίος να τρέχει σε χρόνο $O(|E| \log |V|)$

Πρόβλημα 8

Η Αλίκη θέλει να κάνει ένα πάρτι και προσπαθεί να αποφασίσει ποιους θα καλέσει. Έχει να διαλέξει ανάμεσα σε n ανθρώπους, και έχει κάνει μια λίστα με ζευγάρια ατόμων που γνωρίζονται μεταξύ τους. Θέλει να καλέσει όσον το δυνατόν περισσότερους, αλλά με κάποιους περιορισμούς. Στο πάρτι κάθε καλεσμένος θα πρέπει να γνωρίζεται με τουλάχιστον πέντε άλλους καλεσμένους, και θα πρέπει να μην γνωρίζει τουλάχιστον πέντε άλλους καλεσμένους.

Γράψτε έναν αποτελεσματικό αλγόριθμο ο οποίος παίρνει σαν είσοδο την λίστα των n ατόμων και τη λίστα των ζευγαριών που γνωρίζονται μεταξύ τους και επιστρέφει μια βέλτιστη επιλογή καλεσμένων. Ποιος είναι ο χρόνος που χρειάζεται ο αλγόριθμος σαν συνάρτηση του n ;