Εθνικό Μετσόβιο Πολυτεχνείο
Αλγόριθμοι και Πολυπλοκότητα 7ου εξαμήνου ΣΗΜΜΥ & ΣΕΜΦΕ
http://www.corelab.ece.ntua.gr/courses/algorithms/
Ε. Ζάχος, Α. Παγουρτζής (Τομέας Computer Science ΣΗΜΜΥ)

9η Σειρά Ασκήσεων

(*) Άσκηση 1

Έστω G ένας συνδεδεμένος γράφος με σύνολο κόμβων $V = \{v_1, v_2, \ldots, v_n\}$, που περιλαμβάνει m ακμές και t τρίγωνα (κύκλους μήκους 3).

a. Αν $\mathbf{A}$ είναι ο πίνακας γειτνίασης του γράφου G, αποδείξτε ότι το πλήθος των δρόμων μήκους 2 από τον κόμβο v_i στο v_j ισούται με το στοιχείο που βρίσκεται στη θέση (i, j) του πίνακα $\mathbf{A}^2$.

b. Αποδείξτε ότι το άθροισμα των στοιχείων της κύριας διαγωνίου του πίνακα $\mathbf{A}^2$ ισούται με $2m$.

c. Βρείτε έναν τρόπο να εκφράσετε το πλήθος των δρόμων μήκους 3 από το v_i στο v_j, και συμπεράνετε ότι το άθροισμα των στοιχείων της κύριας διαγωνίου του πίνακα $\mathbf{A}^3$ ισούται με $6t$.

(*) Άσκηση 2

Στο πρόβλημα χρονοπρογραμματισμού διεργασιών έχουμε τις ακόλουθες εισόδους:

- Ένα σύνολο $S = \{1, 2, \ldots, n\}$ από n διεργασίες, κάθε μία από τις οποίες απαιτεί μία χρονική μονάδα για την ολοκλήρωσή της σε έναν επεξεργαστή.
- Ένα σύνολο n ακεραίων deadlines $d_1, d_2, \ldots, d_n$ που ικανοποιούν τη σχέση $1 \le d_i \le n$, για κάθε $1 \le i \le n$. Κάθε διεργασία i υποτίθεται ότι πρέπει να έχει ολοκληρωθεί μέχρι τη χρονική στιγμή d_i.
- Ένα σύνολο n θετικών ακεραίων βαρών $w_1, w_2, \ldots, w_n$. Κάθε διεργασία i που δεν έχει ολοκληρωθεί μέχρι τη χρονική στιγμή d_i προκαλεί κόστος w_i, αλλιώς το κόστος είναι 0.

Ζητείται μία μετάθεση των n διεργασιών, που καθορίζει τη σειρά με την οποία θα εκτελεστούν σε έναν επεξεργαστή. Πρέπει να ελαχιστοποιείται το κόστος που προκαλούν οι καθυστερημένες διεργασίες. Υποτίθεται ότι η εκτέλεση της πρώτης διεργασίας ξεκινάει τη χρονική στιγμή 0 και ολοκληρώνεται τη χρονική στιγμή 1, η εκτέλεση της δεύτερης διεργασίας ξεκινάει τη χρονική στιγμή 1 και ολοκληρώνεται τη χρονική στιγμή 2, κλπ.

Ένας άπληστος αλγόριθμος για το πρόβλημα είναι ο ακόλουθος: εξετάζουμε τις διεργασίες κατά σειρά φθίνοντος βάρους. Κάθε διεργασία που εξετάζουμε την τοποθετούμε όσο πιο αργά γίνεται στο πρόγραμμα, έτσι ώστε ο χρόνος ολοκλήρωσής της να μην ξεπερνάει το αντίστοιχο deadline. Αν αυτό δεν είναι εφικτό, την τοποθετούμε όσο πιο αργά γίνεται στο πρόγραμμα.

Για παράδειγμα, ας υποθέσουμε ότι έχουμε συνολικά 10 διεργασίες και έστω ότι έχουν ήδη δρομολογηθεί δύο διεργασίες που ξεκινούν η πρώτη τη χρονική στιγμή 0 και η δεύτερη τη χρονική στιγμή 1. Η τρίτη διεργασία που εξετάζουμε έχει deadline 7, άρα την προγραμματίζουμε να ξεκινήσει τη χρονική στιγμή 6. Η επόμενη διεργασία που εξετάζουμε έχει deadline 2. Δεν μπορούμε να την προγραμματίσουμε έτσι ώστε να ολοκληρωθεί πριν παρέλθει το deadline, γιατί οι χρονικές στιγμές 0 και 1 είναι ήδη κατειλημμένες. Άρα την τοποθετούμε όσο πιο αργά μπορούμε, τη χρονική στιγμή 9.

Αποδείξτε ότι ο παραπάνω αλγόριθμος δίνει τη βέλτιστη λύση για οποιαδήποτε είσοδο.

(*) Άσκηση 3

Δίνονται δύο ακολουθίες από σύμβολα S_1 και S_2 μήκους n και m αντίστοιχα (στη βιολογία τα σύμβολα αυτά είναι συνήθως τα νουκλεοτίδια A, G, C, T) καθώς και μία συνάρτηση κόστους η οποία για κάθε ζεύγος συμβόλων μας επιστρέφει έναν αριθμό. Το ζητούμενο είναι να παρεμβάλουμε σε κατάλληλες θέσεις στις S_1 και S_2 το σύμβολο "$-$" που συμβολίζει το κενό, έτσι ώστε το άθροισμα του κόστους για τα ζεύγη των συμβόλων που βρίσκονται σε αντίστοιχες θέσεις στις S_1,S_2 να είναι μέγιστο. Το κόστος όταν το ένα από τα δύο σύμβολα είναι το "$-$" είναι σταθερό και συμβολίζεται με d.

Για παράδειγμα, έστω πως έχουμε τα τέσσερα σύμβολα A, G, C, T και τη συνάρτηση κόστους που δίνεται από τον πίνακα

	A	G	C	T
A	10	-1	-3	-4
G	-1	7	-5	-3
C	-3	-5	9	0
T	-4	-3	0	8

Επίσης έστω πως $d = -1$. Τότε μία πιθανή ευθυγράμμιση των $ATCT$ και $TACT$ είναι

$$\begin{array}{cccccc} A & - & T & - & C & T \\ - & T & - & A & C & T \end{array}$$

η οποία δίνει κόστος $4d + 9 + 8 = 13$. Αντιθέτως η προφανής ευθυγράμμιση

$$\begin{array}{cccc} A & T & C & T \\ T & A & C & T \end{array}$$

δίνει κόστος 9.

Χρησιμοποιώντας δυναμικό προγραμματισμό, βρείτε αλγόριθμο που λύνει αυτό το πρόβλημα σε χρόνο πολυωνυμικό ως προς m και n.

(*) Άσκηση 4

Το πρόβλημα 3COLORING είναι το πρόβλημα απόφασης για το αν οι κόμβοι ενός γράφου μπορούν να χρωματιστούν με 3 χρώματα έτσι ώστε να μην υπάρχουν δύο γειτονικοί κόμβοι με το ίδιο χρώμα. Το πρόβλημα NAESAT (not-all-equal SAT) είναι μια παραλλαγή του 3SAT. Παρατηρήστε ότι στο πρόβλημα 3SAT ρωτάμε αν υπάρχει αποτίμηση που για κάθε clause δεν κάνει όλα τα literals της false. Στο πρόβλημα NAESAT ρωτάμε αν για μία φόρμουλα σε μορφή 3-CNF υπάρχει αποτίμηση που την ικανοποιεί με τον περιορισμό για κάθε clause να μην αποτιμά όλα τα literals σε true (με άλλα λόγια ρωτάμε αν υπάρχει αποτίμηση που για κάθε clause να μην αποτιμά τα literals της με τον ίδιο τρόπο). Δεδομένου ότι το πρόβλημα NAESAT είναι **NP**-πλήρες δείξτε ότι και το πρόβλημα 3COLORING είναι **NP**-πλήρες.

Υπόδειξη: Για κάθε μεταβλητή x_i που εμφανίζεται στο instance του NAESAT κατασκευάστε ένα τρίγωνο που θα αντιστοιχεί στους κόμβους x_i, $\neg x_i$, a, όπου a κοινός κόμβος όλων των τριγώνων φτιάχνοντας έτσι ένα αρχικό γράφο. Για κάθε clause του instance του NAESAT κατασκευάστε κατάλληλο γράφο και συνδέστε τον κατάλληλα με τον αρχικό γράφο.

Να παραδοθούν την ημέρα της τελικής εξέτασης, ΠΡΙΝ την εξέταση