ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΑΛΓΟΡΙΘΜΟΙ ΚΑΙ ΠΟΛΥΠΛΟΚΟΤΗΤΑ

3η ΣΕΙΡΑ ΓΡΑΠΤΩΝ ΑΣΚΗΣΕΩΝ

Διδάσκοντες:

Άρ. Παγουρτζής

Δ. Φωτάκης

Δ. Σούλιου

Π. Γροντάς

Ειρήνη Δόντη

ΑΜ 03119839

7ο εξάμηνο

Αθήνα 2023

<u>Άσκηση 1: Υπολογισιμότητα</u>

(a)

Ένας αλγόριθμος που ημιποφασίζει το πρόβλημα των Διοφαντικών Εξισώσεων είναι ο αλγόριθμος απλοποίησης από τη θεωρία των αριθμών, ο οποίος βασίζεται στη χρήση συναρτήσεων απλοποίησης για να απλοποιήσει την εξίσωση και να δείξει ότι η εξίσωση έχει ακέραιες λύσεις αν και μόνο αν η απλοποιημένη εξίσωση έχει ακέραιες λύσεις (δε δίνει συγκεκριμένη λύση).

(b)

Θεωρούμε δεδομένη μία πολυωνυμική εξίσωση με ακέραιους συντελεστές και πεπερασμένο αριθμό αγνώστων και τη μετατρέπουμε σε Διοφαντικό σύστημα εξισώσεων. Χρησιμοποιούμε μία διαδικασία απόφασης, όπως ο αλγόριθμος εξάλειψης ποσοτικοποιητή, για να αποφασίσουμε αν το σύστημα έχει ακέραιες λύσεις. Η διαδικασία απόφασης μπορεί να αναχθεί στο πρόβλημα τερματισμού, κωδικοποιώντας την είσοδο σε μηχανή Turing και ελέγχοντας αν αυτή η μηχανή σταματά σε συγκεκριμένη είσοδο. Το αποτέλεσμα της εκτέλεσης της μηχανής Turing ερμηνεύεται, λοιπόν, ως απάντηση για το πρόβλημα Διοφαντικών Εξισώσεων. Εάν η μηχανή Turing τερματίσει και επιστρέψει "ΝΑΙ," τότε το αρχικό Διοφαντικό σύστημα έχει ακέραιες λύσεις, αλλιώς δεν έχει. Επομένως, αν υπάρξει μία λύση για το πρόβλημα Τερματισμού, μπορεί να λυθεί το πρόβλημα Διοφαντικών Εξισώσεων σε πολυωνυμικό χρόνο.

<u>Άσκηση 2: Πολυπλοκότητα - Αναγωγές</u>

(a)

Εάν ένα πρόβλημα είναι C-πλήρες σε μία σχέση με μία αναγωγή $\leq_R$, τότε το συμπληρωματικό πρόβλημα Π’ είναι επίσης C-πλήρες ως προς το $\leq_R$ και ανήκει στο coC. Οπότε, το Π’ είναι τουλάχιστον τόσο δύσκολο όσο οποιοδήποτε άλλο πρόβλημα στην κατηγορία C και μπορεί να αναχθεί σε Π’ σε πολυωνυμικό χρόνο χρησιμοποιώντας την αναγωγή $\leq_R$. Εφόσον Π' και Π είναι C-πλήρη, τότε το Π' ανήκει στο coC-πλήρες ως προς την αναγωγή.

(b)

Θα δείξουμε ότι SAT $\leq_R$ non-Tautology. Αρχικά, ισχύει ότι η non-Tautology ανήκει στο NP καθώς είναι τουλάχιστον τόσο δύσκολο όσο κάποιο άλλο πρόβλημα στο NP. Επίσης, η non-Tautology είναι ικανοποιήσιμη. Οπότε, η non-Tautology είναι NP-complete και συνεπώς η Tautology είναι coNP-complete.

(c)

Αναγάγουμε όλα τα προβλήματα των NP στο συγκεκριμένο, άρα όλα τα προβλήματα στο NP ανήκουν στο coNP. Το ίδιο συμβαίνει και για το συμπληρωματικό, βάσει του ερωτήματος (α), οπότε αποδεικνύουμε ότι και όλα τα coNP ανήκουν στο NP και συνεπώς το NP είναι το coNP.

(d)

Θα δείξουμε ότι 3SAT $\leq_R$ NAE3SAT. Θεωρούμε την πρόταση $F = C1 \land C2 \land \ldots \land Ck$ 3-φόρμουλα. Θα το μετατρέψουμε σε διαφορετική 3-φόρμουλα G η οποία είναι NAESAT αν και μόνο αν η F είναι SAT. Θεωρούμε καινούριες μεταβλητές z, w1,..., wk. $Cj = \lambda 1 \lor \lambda 2 \lor \lambda 3$ και $Dj = (\lambda 1 \lor \lambda 2 \lor wj) \land (\neg wj \lor \lambda 3 \lor z)$ με $G = D1 \land D2 \land \ldots \land Dk$ (2k όροι). Ο μετασχηματισμός απαιτεί πολυωνυμικό χρόνο. Η F είναι ικανοποιήσιμη:

Αν $\lambda 1 \lor \lambda 2 = 1$, με wj=0, z=0 ή αν $\lambda 1 \lor \lambda 2 = 0$, με wj=1, z=0, τότε και η G είναι NAESAT. Δηλαδή αν (x1,x2,...,xn,w1,...,wk,z=1), NAE-ικανοποιεί την G αν και μόνο αν $(\neg x1,\ldots,\neg xn,\neg w1,\ldots,\neg wk,z=0)$ την ικανοποιεί. Υποθέτοντας ότι z = 0, τότε Dj είναι NAESAT αν και μόνο αν $\lambda 1 \lor \lambda 2 = 1$ or $\lambda 3 = 1$, οπότε Cj είναι ικανοποιήσιμη.

(e)

Μπορούμε να επιβεβαιώσουμε ότι το πρόβλημα Επιλογής Αντιπροσώπων είναι NP-complete μέσω της αντιστοιχίας σε Set Cover πρόβλημα. Για το Set Cover Θεωρούμε σύνολο S, υποσύνολα x1,…,xm του S και 1<k<m. Υπάρχουν το πολύ k υποσύνολα που η ένωση τους είναι το S. Παρατηρούμε ότι το πρόβλημα Επιλογής Αντιπροσώπων είναι

ουσιαστικά ένα Set Cover πρόβλημα για το οποίο γνωρίζουμε ότι είναι NP-πλήρες. Οπότε και το πρόβλημα Επιλογής Αντιπροσώπων είναι NP-πλήρες.

Άσκηση 3: Προσεγγιστικοί Αλγόριθμοι – Vertex Cover

Ο αλγόριθμος είναι ένας αλγόριθμος 2 προσεγγίσεων για το πρόβλημα του Vertex Cover. Συγκεκριμένα, **το** σύνολο **C** που επιστρέφεται από τον αλγόριθμο **καλύπτει όλες τις ακμές** στο γράφημα G. Αυτό συμβαίνει, καθώς ο αλγόριθμος επιλέγει επανειλημμένα μία ακάλυπτη ακμή και προσθέτει τουλάχιστον ένα από τα τελικά σημεία του στο C μέχρι να καλυφθούν όλες οι ακμές. **Το συνολικό βάρος του συνόλου C που επιστρέφεται δεν είναι περισσότερο από το διπλάσιο του βάρους του βέλτιστου καλύμματος κορυφής**, καθώς ο αλγόριθμος εκχωρεί ένα βάρος c(e) σε κάθε ακμή e, το οποίο είναι ίσο με το ελάχιστο των βαρών των τελικών σημείων της ακμής και το συνολικό βάρος του C είναι το άθροισμα αυτών των βαρών για όλες τις ακμές. Εφόσον σε κάθε ακμή εκχωρείται ένα βάρος που δεν υπερβαίνει το μισό βάρος των τελικών σημείων του, το συνολικό βάρος του C δεν είναι μεγαλύτερο από το διπλάσιο του βάρους βέλτιστου καλύμματος κορυφής. **Το συνολικό βάρος του καλύμματος βέλτιστης κορυφής δεν είναι μικρότερο από το συνολικό βάρος του C**, επειδή ο αλγόριθμος μειώνει μόνο τα βάρη των κορυφών. Οπότε, το βάρος του C είναι το χαμηλότερο όριο στο βάρος του βέλτιστου καλύμματος κορυφής.

Ένα παράδειγμα γράφου στο οποίο ο αλγόριθμος υπολογίζει ένα Vertex Cover με συνολικό βάρος διπλάσιο από αυτό της βέλτιστης λύσης είναι ένα γράφημα με δύο κορυφές και μία ακμή. Η μία κορυφή πρέπει να έχει βάρος 1 και η άλλη να έχει βάρος 2. Οπότε, ο αλγόριθμος θα επέλεγε την κορυφή με βάρος 1 και θα επέστρεφε ένα κάλυμμα κορυφής βάρους 1, ενώ το βέλτιστο κάλυμμα θα ήταν η κορυφή με βάρος δύο.

Άσκηση 4: Πιθανοτικοί Αλγόριθμοι – Έλεγχος Ταξινόμησης

(α)

Επιλέγονται τυχαία k θέσεις στον πίνακα και ελέγχουμε αν τα στοιχεία σε αυτές τις θέσεις είναι στη σωστή σειρά. Ο αλγόριθμος μπορεί να αποφανθεί στην περίπτωση πινάκων που δεν είναι σχεδόν ταξινομημένοι, αλλά έχουν μικρό αριθμό τυχαία τοποθετημένων στοιχείων που ικανοποιούν το κριτήριο. Ένα παράδειγμα πίνακα που θα απαιτήσει **Ω**(n) ελέγχους και θα έχει πιθανότητα λάθους μικρότερη του 10% μπορεί να είναι ο εξής: A = [1, 2, 3, 7, 4, 5, 6]. Σε αυτόν τον πίνακα, ο αλγόριθμος θα πρέπει να ελέγξει τις θέσεις 7, 4 και 5 για να διαπιστώσει ότι ο πίνακας δεν είναι σχεδόν ταξινομημένος. Αυτός ο έλεγχος απαιτεί **Ω**(n) ελέγχους, καθώς για κάθε θέση του πίνακα μπορεί να χρειαστεί να εξετάσει μια τυχαία σειρά θέσεων για την πραγματοποίηση του ελέγχου. Αν ο αλγόριθμος επιλέγει αυτές τις θέσεις για έλεγχο, η πιθανότητα να κάνει λάθος σε αυτόν τον πίνακα θα είναι μικρή, καθώς οι τρεις αυτές θέσεις είναι εκείνες που διαφέρουν από τον τυπικό ταξινομημένο πίνακα.

(β)

Η δυαδική αναζήτηση χωρίζει επανειλημμένα το διάστημα αναζήτησης στη μέση, απορρίπτοντας το μισό μέρος που δεν περιέχει το στοιχείο στόχο. Όταν καλείται η BINARY-SEARCH(A, x, 1, n) και επιστρέφει τη θέση k, τότε το διάστημα αναζήτησης έχει περιοριστεί στο διάστημα [k,k] και το x βρίσκεται στη θέση k. Επίσης, όταν καλείται η BINARY-SEARCH(A, y, 1, n) και επιστρέφει τη θέση ℓ, τότε το διάστημα αναζήτησης έχει περιοριστεί στο διάστημα [ℓ, ℓ] και το y βρίσκεται στη θέση ℓ. Εφόσον k < ℓ, σημαίνει ότι το x βρέθηκε πριν τη διαδικασία αναζήτησης, κάτι που μπορεί να συμβεί μόνο αν x < y. Οπότε, αν k < ℓ, τότε x < y.

(γ)

Πρέπει να αποδείξουμε ότι αν ένας πίνακας A δεν είναι σχεδόν ταξινομημένος, τότε με μεγάλη πιθανότητα, σε τουλάχιστον μία από τις τυχαία επιλεγμένες θέσεις a_i, δε θα ικανοποιεί τη συνθήκη ai = BINARY-SEARCH(A, A[a_i], 1, n). Για να αποδειχθεί το ζητούμενο, χρησιμοποιούμε το γεγονός ότι αν το A δεν είναι σχεδόν ταξινομημένος, τότε υπάρχουν τουλάχιστον n/4 στοιχεία που πρέπει να αφαιρεθούν για να ταξινομηθεί το A. Αν επιλέξουμε τυχαία k θέσεις a1, a2, …, ak τότε η πιθανότητα να αντιστοιχούν όλα τα στοιχεία που πρέπει να αφαιρεθούν για να ταξινομηθεί το A είναι $(n/4)^k$. Η παραπάνω πιθανότητα γίνεται 10% για k=4.