Εθνικό Μετσόβιο Πολυτεχνείο
Σχολή ΗΜ&ΜΥ
Αλγόριθμοι και Πολυπλοκότητα
$7^{ο}$ εξάμηνο
Ακαδημαϊκή περίοδος: 2015-2016

$4^{η}$ Σειρά Γραπτών Ασκήσεων

Ψαρουδάκη Ελένη
Α.Μ.: 03112080

16 Φεβρουαρίου 2016

Άσκηση 1: Επιβεβαίωση Συντομότερων Μονοπατιών

Για την λύση αυτού του προβήματος αρκεί να εκτελέσουμε μία αναζήτηση BFS που ξεκινάει από την κορυφή v_1. Έχουμε επιπλόν ένα πίνακα p, αρχικοποιημένο στο μηδέν, όπου κρατάμε τον πατέρα του κάθε κόμβου (πατέρας του πρώτου κόμβου ο κόμβος θεωρείται το 0 καθώς η αρίθμηση των κόμβων ξεκινάει από το 1). Κατά την εκτέλεση της αναζήτησης ελέγχουμε, για κάθε κόμβο i, αν ισχύει η συνθήκη:

$$\delta_i + w_{ij} < \delta_j$$

όπου j οι κόμβοι παιδιά. Αν η συνθήκη δεν ισχύει έχουμε αποδείξει πως οι αποστάσεις δεν ανταποκρίνονται στις αποστάσεις των κορυφών από την κορυφή v_1. Στην αντίθετη περίπτωση συνεχίζουμε, αφού πρώτα αλλάξουμε την τιμή στην θέση του πίνακα ($p[j] = i$). Μετά το τέλος της διαδικασίας, ελέγχουμε αν όλες οι θέσεις του πίνακα έκτός από την 1 έχει τιμή διαφορετική του μηδενός. Αν όχι ο πίνακας αυτός αποτελεί το Δέντρο Συντομότερων Μονοπατιών. Αν ναι οι αποστάσεις θεωρούνται λανθασμένες. Η επίλυση αυτή έχει κόστος $\mathcal{O}(n+m)$, ίσο με το κόστος του BFS.

Άσκηση 2: Τροποποίηση του Αλγόριθμου Dijkstra (DPV 4.17)

(α)

Για την επίλυση της άσκησης θα υλοποιήσουμε τον αλγόριθμο του Dijkstra τροποποιημένο ώστε να χρησιμομποιεί buckets. Με την παρακάτω υλοποίηση θα επιτύχουμε χρόνο εκτέλεσης $\mathcal{O}(nC+m)$, όπου n ο αριθμός των κορυφών και m ο αριθμός των ακμών.

Συγκεκριμένα ομαδοποιούμε τους κόμβους, για τους οποίους δεν έχει υπολογιστεί ακόμα οι τελικές αποστάσεις, σε buckets με κριτήριο την απόσταση που έχουν μέχρι την προσωρινή απόσταση. Οι κόμβοι σε κάθε bucket βρίσκονται σε μία διπλά συνδεδεμένη λίστα. Καθώς έχουμε μέγιστη απόσταση C, δεν θα μπορούν να υπάρχουν περισσότεροι από $C+1$ buckets, καθώς υπάρχει οι αποστάσεις είναι άνω φραγμένες. Βρίσκουμε το bucket που περιέχει τουλάχιστον ένα κόμβο και έχει την μικρότερη ετικέτα. Για όλους τους κόμβους αυτού του bucket μονιμοποιούμε την την απόσταση και τους διαγράφουμε από το bucket. Στην υλοποίηση αυτή ελέγχουμε για κάθε bucket αν είναι άδειο και μπορούμε μόνο να προσθέσουμε και να διαγράψουμε κόμβους από το bucket. Αν αλλάξει η απόσταση ενός κόμβου πρέπει να ανανεώνεται και η προσωρινή ετικέτα του κόμβου. Τέλος η διαδικασία επαναλαμβάνεται μέχρι να αδειάσουν όλα τα buckets.

Καθώς ελέγχουμε όλες τις ακμές υπάρχει κόστος $\mathcal{O}(m)$ και για την εύρεση του επόμενου bucket έχουμε κόστος $\mathcal{O}(C)$, το οποίο επαναλαμβάνεται μέχρι να διαγραφούν όλοι οι κόμβοι n. Έτσι προκύπτει η ζητούμενη πολυπλοκότητα.

(β)

Αφού υπάρχει ο νέος περιορισμός, μπορούμε να προσαρμόσουμε την παραπάνω υλοποιήση. Η μέγιστη απόσταση μας επιτρέπει να χρησιμοποιήσουμε ισοζυγισμένα δέντρα διαδικής αναζήτησης. Με τον τρόπο αυτό έχουμε κόστος αναζήτησης του επόμενου bucket της τάξης $\mathcal{O}(\log(2^C)) = \mathcal{O}(C)$ (ίσο με το μήκος του δέντρου) και η διαδικασία επαναλαμβάνεται για κάθε κόμβο. Επιπλέον έχουμε σε αντίθεση με πριν κόστος $\mathcal{O}(C)$ για την μετακίνηση των ακμών ενός κόμβου στην κορυφή του δέντρου (προηγουμένως είχαμε κόστος O(1)). Συνολικά προκύπτει πολυπλοκότητα $\mathcal{O}((n+m)C)$.

Άσκηση 3: Συντομότερα Μονοπάτια με δύο Κριτήρια

(α)

Το ερώτημα αυτό λύνεται με τη δημιουργία ενός γραφήματος δύο επιπέδων. Θα δημιουργήσουμε ένα αντίγραφο του αρχικού γραφήματος που θα περιέχει μόνο τις πράσινες ακμές. Δημιουργούμε ένα ακόμα αντίγραφο του αρχικού, στο οποίο οι κόκκινες ακμές του αρχικού γραφήματος θα έχουν αντικατασταθεί με κόκκινες γραμμές που συνδέουν τα δύο διαφορετικά αντίγραφα στους αντίστοιχους κόμβους. Θα τρέξουμε τον αλγόριθμο του Dijkstra στο συνολικό γράφημα που προκύπτει, επιτρέποντας μόνο μία μετάβαση από το επίπεδο ένα στο δύο (θεωρούμε πως η αρχική κορυφή είναι μοναδική). Έστω οι αποστάσεις $s-v_0$ και $s-v_1$ (όπου v_0 και v_1 οι αντίστιχοι κόμβοι στα δύο επίπεδα). Από το Dijkstra έχει προκύψει η απόσταση έχοντας μόνο πράσινες ή μία κόκκινη ακμή. Από αυτές τις δύο το min αντιστοιχεί στο shortest path, για τον κόμβο v του αρχικού γραφήματος. Από τις παραπάνω παραδοχές προκύπτει και η ορθότητα του αλγορίθμου καθώς δεν μπορούμε να μεταβούμε από το ένα επίπεδο στο άλλο ελεύθερα (μόνο μία μετάβαση επιτρεπτή). Για τη δημιουργία του νέου γραφήματος απαιτείται $\mathcal{O}(2m) = \mathcal{O}(m)$. Το συνολικό κόστος του αλγόριθμου θα είναι $\mathcal{O}(n \log n + m)$

(β)

Στη γενικευμένη περίπτωση θέλουμε να ισχύει για το κόστος : C <= k. Επομένως χρειαζόμαστε k επίπεδα και επομένως k αντίγραφα του G, στα οποία γίνεται η αντικατάσταση των κόκκινων ακμών, με όμοια συλλογιστική πορεία με το (α) ερώτημα. Το ελάχιστο μονοπάτι θα έχει κόστος το πολύ k καθώς η απόσταση θα είναι το min που θα προκύψει από τις αποστάσεις των αντίστοιχων κορυφών $s - v_0, s - v_1, .., s - v_{k-1}$. Για τη δημιουργία του συνολικού γραφήματος χρειάζεται $\mathcal{O}(km)$ και για τον υπολογισμό των διαδρομών $\mathcal{O}(kn \log kn + km)$. Καθώς το k είναι άνω φραγμένο από το n προκύπτει τελικά πολυπλοκότητα $\mathcal{O}(n^2 \log n + mn)$.

Άσκηση 4: Σύστημα Ανισοτήτων και Μετατροπή Σταθερών Όρων

(α)

Για την επίλυση της άσκησης αναγάγουμε το πρόβλημα σε κάποιο αντίστοιχο με γράφους. Σε κάθε μεταβλητή αντιστοιχεί ένας κόμβος και κάθε περιορισμός μεταξύ των μεταβλητών σε μια ακμή, μεταξύ των αντίστοιχων κόμβων, βάρους ίσου με τον περιορισμό (κόστος δημιουργίας γραφήματος $\mathcal{O}(m)$).

Το σύστημα των ανισοτήτων είναι ικανοποιήσιμο αν και μόνο αν δεν δεν υπάρχει κάποιος αρνητικός κύκλος στο γράφημα. Θα εφαρμόσουμε τον αλγόριθμο του Bellman-Ford ξεκινώντας από τον κόμβο S, ο οποίος είναι συνδεδεμένος με όλους τους υπόλοιπους κόμβους με ακμές βάρους ίσες με 0. Σε κάθε μεταβλητή c_i δίνουμε τιμή ίση με την ελάχιστη απόσταση που προκύπτει από τον αρχικό κόμβο S. Αν υπάρχει αρνητικός κύκλος, το σύστημα δεν είναι ικανοποιήσιμο καθώς θα προκύψει τουλάχιστον μία διαδρομή όπου η ανισότητα δεν θα ικανοποιείται (συνεχώς θα μειώνεται η απόσταση-αρνητική τιμή). Αν δεν υπάρχει, τότε για τις ακμές που ανήκουν στο δέντρο ελάχιστων διαδρομών ισχύει η ισότητα, ενώ για τις υπόλοιπες η ανισότητα (εξαιτίας της τριγωνικής ανισότητας). Η πολυπλοκότητα θα είναι $\mathcal{O}(nm)$ (κόστος τους αλγορίθμου).

(β)

Αν το S είναι μη ικανοποήσιμο, τότε δεν μπορεί να υπάρχει ικανοποιήσιμο S' με θετικά βάρη (ο έλεγχος γίνεται με τον αλγόριθμο που περιγράφηκε στο (α) ερώτημα, από όπου προκύπτουν και τα κόστη). Αν είναι ικανοποιήσιμο, χρησιμοποιούμε την ιδέα του αλγορίθμου του Johnson. Χρησιμοποιούμε για κάθε κόμβο την συνάρτηση Change(S,i,-c_i), καθώς για ακμή που εισέρχεται στον κόμβο θέλουμε να αφαιρέσουμε b_{ij}, ενώ για κάθε ακμή που εξέρχεται να προσθέσουμε b_{li}. Τα βάρη με αυτόν τον τρόπο παραμένουν θετικά επομένως προκύπει πως αν το S ήταν ικανοποιήσιμο και το S' είναι ικανοποιήσιμο. Η χρονική πολυπλοκότητα θα είναι $\mathcal{O}(nm + 2m)$

Άσκηση 5: Διαφημίσεις στο Διαδύκτιο

Για την επίλυση της άσκησης θα δημιουργήσουμε ένα αντίστοιχο πρόβλημα με γράφους. Θα υπάρχουν n κόμβοι για τους ανθρώπους (έστω a_i), k κόμβοι για τις κατηγορίες (έστω b_i) και m κόμβοι για τις εταιρίες (έστω c_i). Έστω επίσης αρχικός κόμβος s και τελικός f (ο αρχικός έχει εξαρχόμενες ακμές σε όλες τις εταιρίες με κόστος ίσο με τον αριθμό των διαφημίσεων. Ο τελικός έχει εισερχόμενες ακμές με κόστος 1 από κάθε επισκέπτη). Καθώς κάθε κατηγορία ενώνεται με τους ανθρώπους θεωρούμε πως υπάρχουν εξαρχόμενες ακμές με κόστος 1 σε κάθε άνθρωπο που ανήκει στην κατηγορία και εισερχόμενες στην κατηγορία από τις εταιρίες με κόστος 0 (ο ρόλος των κόμβων των κατηγοριών είναι μεταβατικός, μπορούν να αφιρεθούν τελείως από το πρόβλημα). Λύνουμε το πρόβλημα με χρήση της μεθόδου Ford-Fulkerson για max flow από το s στο f. Αν κάποια εταιρία δεν έχει μεταβατικά εξαρχόμενες ακμές (αθροιστικό κόστος μγαλύτερο του μηδενός) δεν μπορεί να λυθεί το πρόβλημα. Σε αντίθετη περίπτωση θα προβληθούν x διαφημίσεις σε x ανθρώπους αρκεί να υπάρχουν ακμές.

Άσκηση 6: Αναγωγές και NP-Πληρότητα

Υποσύνολα Διαφορετικών Συνόλων

Το πρόβλημα ανήκει στο NP.

Το πρόβλημα αναγάγεται πολυωνυμικά από το Subset Sum. Έστω σύνολο A και αποσύνολο S με βάρος W, τότε αν λύσουμε το πρόβλημα X=A, Y=1 και B=W-1 θα υπάρχει λύση όπου τελικά S=S_1 και S_2=Y.

Πιστοποιητικό: Μπορούμε να υπολογίσουμε σε γραμμικό χρόνο τα αθροίσματα αν μας δώσουν τα υποσύνολα S_1, S_2

Άθροισμα Υποσυνόλου κατά Προσέγγιση

Το πρόβλημα ανήκει στο NP.

Αναγάγουμε το Subset Sum στο πρόβλημα μας. Δεδομένου στιγμιότυπου A=$w_1, w_2, ..., w_n$ του Subset Sum αν W το βάρος, τότε αν A'=$2w_1, 2w_2, ...2w_n$ θα έχει βάρος 2W. και x=1. Όλες οι τιμές του A' είναι άρτιες επομένως, αν υπάρχει λύση το βάρος θα είναι απαραίτητα άρτιο.

Πιστοποιητικό: Μπορούμε να επαληθεύσουμε σε γραμμικό χρόνο, πως το δεδομένο σύνολο είναι η λύση.

Μακρύ Μονοπάτι

Το πρόβλημα ανήκει στο NP.

Για να το αποδείξουμε αναγάγουμε το hamiltonian path σε αυτό. Στο γράφο G, n κορυφών του hamiltonian path, προσθέτουμε n απομονωμένες κορυφές. Το νέο γράφημα (έστω G') περιέχει το G με και χωρίς ακμές. Το μονοπάτι που ενώνει τις μισές κορυφές στον G' θα είναι hamiltonian path στον G'.

Πιστοποιητικό: Το ίδιο το μονοπάτι.

Επιλογή Ανεξάρτητων Υποσυνόλων

Το πρόβλημα ανήκει στο NP.

Πιστοποιητικό: Αν μας δώσουν την διαμέριση σε υποσύνολα μπορούμε να ελέγξουμε σε χρόνο $\mathcal{O}(n^2)$ αν ικανοποιούνται οι περιορισμοί.

Διαχωρισμός σε Κλάσεις

Το πρόβλημα ανήκει στο NP.

Το πρόβλημα αναγάγεται πολυωνυμικά από το 3-Coloring. Αν οι κορυφές είναι γειτονικές τότε η απόσταση ισούται με d, αν δεν είναι με 0. Θέτουμε B=d-1 και k=3. Αποδεκτά, για το 3-Coloring, είναι μόνο τα στιγμιότυπα όπου κάθε σύνολο έχει μόνο σημεία με μηδενικές αποστάσεις (γειτονικές κορυφές) και κατα συνέπεια αποδεκτά και για το Διαχωρισμό σε κλάσεις.

Πιστοποιητικό: Μπορούμε να επαληθεύσουμε τις αποστάσεις που χρειάζονται σε πολυωνυμικό χρόνο.

Συνεκτικό Δέντρο με Περιορισμούς

Το πρόβλημα ανήκει στο NP.

Μπορεί να αναγχθεί το πρόβλημα του knapsack πολυωνυμικά σε αυτό. Μας δίνεται η χωρητικότητα (W) και ο ζητούμενη συνολική αξία (P). Οι ακμές αποτελούν τα αντικείμενα που θα βάλουμε στο σάκο μας.

Πιστοποιητικό: Το ίδιο το δέντρο. Αν στο δώσουν μπορείς να υπολογίσεις τα κόστη και να ελέγξεις τη τήρηση των περιορισμών.