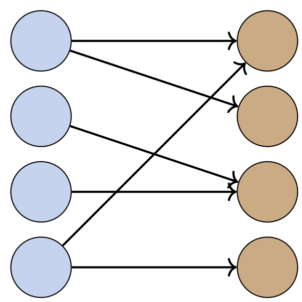
Δημήτρης Κοκκίνης 03118896
Ιανουάριος, 2024

Αλγόριθμοι και Πολυπλοκότητα
3η Σειρά Γραπτών Ασκήσεων

Άσκηση 1: Διαφημίσεις στο Διαδίκτυο

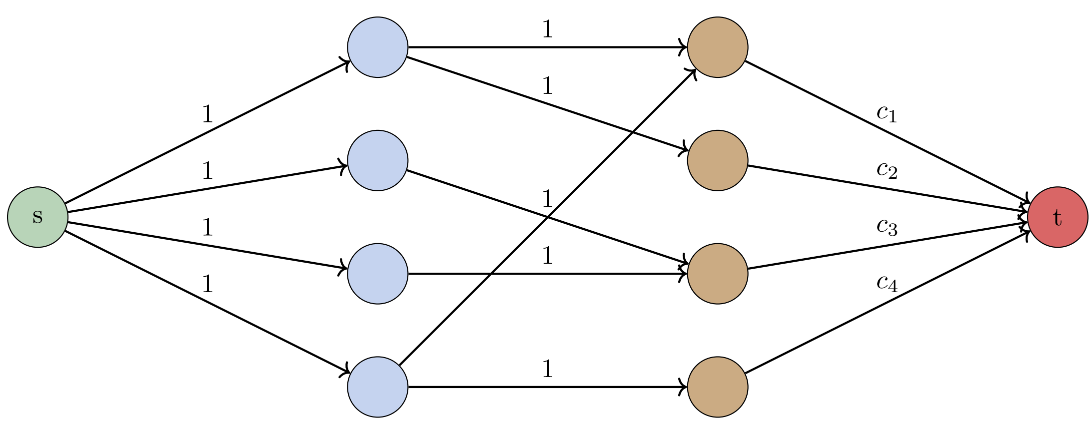
Απο τη περιγραφή του προβλήματος, παρατηρούμε μια διμερή σχέση ανάμεσα στις εταιρείες και στους επισκέπτες.



Οι μπλε κόμβοι είναι οι επισκέπτες, οι οποίοι συνδέονται με τις διαφημιστικές εταιρείες της πορτοκαλί στήλης αν και μόνο αν ανήκουν στις κατηγορίες στις οποίες θέλει να προβάλλονται οι διαφημίσεις της η εκάστοτε εταιρεία.

Αυτό μας παραπέμπει στην αναγωγή του προβλήματος στο πρόβλημα εύρεσης μέγιστου ταιριάσματος, το οποίο όπως γνωρίζουμε μπορεί να λυθεί κατασκευάζοντας το αντίστοιχο γράφημα και βρίσκοντας τη μέγιστη ροή. Μάλιστα, οι δύο παρακάτω προτάσεις της εκφώνησης παρέχουν τη πληροφορία που είναι απαραίτητη για τη κατασκευή του γραφήματος.

- ”Εμφανίζεται μια διαφήμιση σε κάθε επισκέπτη”. Επομένως, οι χωρητικότητα των ακμών προς τους κόμβους των επισκεπτών θα είναι 1
- ”Προβάλλονται το πολύ c_i διαφημίσεις κάθε εταιρείας i”, το οποίο υποδυκνύει τη χωρητικότητα των ακμών με αρχή τους κόμβους των εταιρειών ως c_i



Επομένως, θα κατασκευάσουμε ένα γράφημα, με ένα κόμβο πηγή s, ένα κόμβο για κάθε επισκέπτη $j = (1, ..., n)$, ένα κόμβο για κάθε εταιρεία $i = (1, ..., m)$ και ένα καταληκτικό κόμβο t. Οι ακμές απο τη πηγή μέχρι τους κόμβους των επισκεπτών θα έχουν χωρητικότητα 1 καθώς οι επισκέπτες μπορούν να δουν μια διαφήμιση. Η ακμή (j,i) που συνδέει έναν επισκέπτει j με μια εταιρεία i θα έχει χωρητικότητα 1 αν ο επισκέπτης ανήκει σε μια απο τις δημογραφικές ομάδες του συνόλου S_i της εταιρείας i. Τέλος, οι ακμές απο κάθε κόμβο εταιρείας προς το καταληκτικό κόμβο t θα έχουν χωρητικότητα c_i.

Χρησιμοποιώντας τον αλγόριθμο Edmonds-Karp, μπορούμε να βρούμε μια μέγιστη ροή απο τον s στον t. Αν στο τέλος η μέγιστη ροή είναι n τότε είναι δυνατό να εμφανιστεί μια διαφήμιση σε κάθε επισκέπτη με τους περιορισμούς των εταιρειών.

Algorithm

1. Construct graph G

    Initialize node s and t

    for each visitor j=1 **to** n

        add edge (s, v_j) with capacity 1

    for each company i=1 **to** m

        add edge (v_i, t) with capacity c[i]

    for each visitor j=1 **to** n

        for each company i=1 **to** m

            for each category k in S[i]

                if k **isin** categories[j] then add edge (v_j, v_i) with capacity 1

2. Find max flow

3. **if** max flow == n

    print("Possible")

    for each visitor j=1 **to** n

        for each company i=1 **to** m

            if (v_j, v_i) == 1 **exists**

                print("Visitor j watches ad of company j")

Για τη κατασκευή του δικτύου χρειαζόμαστε χρόνο $\mathcal{O}(n * m * k)$. Ο Edmonds-Karp κάνει χρόνο $\mathcal{O}(V * E^2)$, όπου στη περίπτωσή μας έχουμε n+m+2 κόμβους και μικρότερες απο n*m ακμές. Άρα η πολυπλοκότητα του Edmonds-Karp θα είναι $\mathcal{O}((n + m) * (n * m)^2)$.

Άσκηση 2: Κυκλικές συμβολοσειρές

Η κυκλική συμβολοσειρά είναι απλά μια θεωρητικά άπειρη επανάληψη της αρχικής συμβολοσειράς p2. Έτσι, αυτό που χρειάζεται να κάνουμε είναι να ελέγξουμε αν η p1 είναι υποσυμβολοσειρά κάποιας κυκλικής μετατόπισης της p2. Η αναζήτηση της υποσυμβολοσειράς μπορεί να υλοποιηθεί με τον αλγόριθμο Rabin-Karp με πολυπλοκότητα $\mathcal{O}(n1 + n2)$, όπου n1 το μήκος της συμβολοσειράς p1 και n2 το μήκος της συμβολοσειράς p2. Αν δεν βρεθεί καμιά μετατόπιση, τότε η p1 δεν είναι υποσυμβολοσειρά της κυκλικής p2.

Άσκηση 3: Huffman coding

(a)

Ο κώδικας είναι εφικτός καθώς καμία κωδικολέξη δεν είναι prefix μιας άλλης. Ένα παράδειγμα συχνοτήτων είναι να αναθέσουμε στο χαρακτήρα με την υψηλότερη συχνότητα εμφάνισης το μικρότερο κώδικα και οι άλλοι δύο χαρακτήρες να πάρουν μεγαλύτερους κώδικες. Πχ $f_a = 0.5, f_b = 0.25, f_c = 0.25$

Ο αλγόριθμος Huffman πρώτα θα συνδυάσει τα δύο σύμβολα με τη μικρότερη συχνότητα σε ένα κόμβο. Ύστερα, οι υπόλοιποι κόμβοι θα συνδυαστούν για να σχηματίσουν τη ρίζα, με αποτέλεσμα να έχουμε τη κωδικοποίηση: $a \to 0, b \to 10, c \to 11$

(b)

Ο συγκεκριμένος κώδικας είναι αδύνατος, διότι το '0' είναι πρόθεμα του '00', κάτι που δεν μπορεί να προκύψει απο τον αλγόριθμο Huffman. Αυτή η κωδικοποίηση μπορεί να οδηγήσει σε διφορούμενες συμβολοσειρές.

(c)

Ο αλγόριθμος Huffman δεν μπορεί να παράγει το κώδικα αυτό για οποιεσδήποτε συχνότητες, καθώς ο αλγόριθμος αναθέτει μικρότερους κωδικούς στα σύμβολα με υψηλότερες συχνότητες. Στη περίπτωση αυτή όμως, όλοι οι κωδικοί έχουν το ίδιο μήκος που σημαίνει πως όλες οι συχνότητες είναι ίδιες. Ωστόσο, με ίσες συχνότητες ο αλγόριθμος Huffman θα ανέθετε κωδικούς με τη σειρά 00, 01, 10 και όχι 10, 01, 00.

Άσκηση 4: Υπολογισιμότητα

(a)

Για να αποδείξουμε πως το πρόβλημα διοφαντικών εξισώσεων ανήκει στην κλάση RE, θα δείξουμε πως υπάρχει μηχανή Turing η οποία θα τερματίζει αν υπάρχει λύση, αλλιώς θα τρέχει επ'άπειρον αν δεν υπάρχει λύση. Συγκεκριμένα, η μηχανή Turing θα απαριθμεί όλους του πιθανούς ακέραιους $x_1, x_2, ..., x_n$ και θα ελέγχει αν κάποιος απο αυτούς ικανοποιεί την διοφαντική εξίσωση.

Η είσοδος στη μηχανή θα είναι η δεδομένη διοφαντική εξίσωση. Στη συνέχεια, η μηχανή θα ξεκινάει απο το (0, 0, ..., 0) και θα συνεχίζει. (1, 0, ..., 0), (0, 1, ..., 0), ..., και ούτω κάθε εξής. Σε κάθε βήμα θα υπολογίζει το $P(x_1, x_2, ..., x_n)$ που είναι η εξίσωση. Αν $P(x_1, x_2, ..., x_n) = 0$ θα τερματίζει αλλιώς θα συνεχίζει.

Το πρόβλημα διοφαντικών εξισώσεων ανήκει στη κλάση των ημιαποκρίσιμων υπολογιστικών προβλημάτων, καθώς υπάρχει μηχανή Turing, η οποία αν βρεθεί λύσει τερματίζει, διαφορετικά δεν τερματίζει.

(b)

Αρχικά θα αποδείξουμε πως το Halting Problem είναι στη κλάση RE. Δοσμένης μηχανής M και εισόδου w, μπορούμε να σχεδιάσουμε μηχανή M' που προσομοιώνει τη μηχανή M(w). Αν η M(w) τερματίζει, η M' τερματίζει και αποδέχεται. Αν η M(w) δεν τερματίσει, η M' τρέχει επ'αόριστον.

Στη συνέχεια, θα αποδείξουμε πως ένα πρόβλημα που ανήκει στη κλάση RE μπορεί να αναχθεί στο HP πρόβλημα. Αυτό σημαίνει, πως για κάθε πρόβλημα L υπάρχει μια υπολογίσιμη συνάρτηση που μετατρέπει στιγμιότυπα του

L σε στιγμιότυπα του HP, τέτοια ώστε ένα στιγμιότυπο του L γίνεται αποδεκτό αν και μόνο αν το αντίστοιχο στιγμιότυπο του HP γίνεται αποδεκτό. Έστω $L \in \mathbf{RE}$. Τότε, θα υπάρχει μηχανή Turing M_L, η οποία θα ημιαποφασίζει την L. Κατασκευάζουμε τώρα τη μηχανή M'_L που προκύπτει απο την αρχική αφαιρώντας κάθε μετάβαση που οδηγεί σε κατάσταση απόρριψης απο τη συνάρτηση μεταβάσεών της. Αν η M_L απορρίπτει σε κάποιο βήμα, η M'_L κολλάει. Θεωρούμε $f : \Sigma^* \to \Sigma^*$. Για $x \in L$ τότε $M_L(x) = YES$ και $M'_L(x) = YES$ άρα η μηχανή $M'_L(x)$ τερματίζει, οπότε $\langle M'_L, x\rangle \in \mathbf{HP}$, άρα $f(x) \in \mathbf{HP}$ επομένως η f είναι αναγωγή απο την L στο HP δηλαδή η L ανάγεται στο HP. Έτσι, συμπεραίνουμε πως το HP είναι RE-hard, επομένως το HP είναι RE-πλήρες.

Άσκηση 5: Πολυπλοκότητα- Αναγωγές

(a)

Αρχικά δείχνουμε πως το πρόβλημα της ταυτολογίας είναι coNP. Αυτό ισχύει διότι το συμπληρωματικό του προβλήματος, δηλαδή δοσμένου τύπου να εξετάσουμε αν δεν είναι ταυτολογία, είναι NP. Έπειτα, αρκεί να δείξουμε πως το Tautology Problem είναι coNP-hard. Για αυτό το σκοπό, θα ανάγουμε οποιοδήποτε πρόβλημα που ανήκει στη κλάση coNP στο πρόβλημα ταυτολογίας. Έστω γλώσσα L στη coNP, τότε η συμπληρωματική της θα ανήκει στην NP. Αναγάγουμε την $\bar{L}$ στο πρόβλημα ικανοποιησιμότητας (SAT), το οποίο είναι NP-complete.

Δοσμένου τύπου ψ, σκοπός είναι να αποφασίσουμε αν το ψ είναι ικανοποιήσιμο. Για να ελέγξουμε την ικανοποιησιμότητα του ψ, αρκεί να ελέγξουμε αν το $\neg\psi$ είναι ταυτολογία. Άρα, εφόσον το SAT είναι NP-complete, αυτή η αναγωγή διασφαλίζει πως το πρόβλημα της ταυτολογίας είναι coNP-hard.

(b)

Για αυτό το πρόβλημα, οι περιπτώσεις "όχι" μπορούν να επαληθευθούν όπως και οι περιπτώσεις "ναι" σε πολυωνυμικό χρόνο. Εφόσον μπορούν να επαληθευτούν τόσο οι θετικές όσο και οι αρνητικές περιπτώσεις σε πολυωνυμικό χρόνο για NP-πλήρες πρόβλημα, τότε η διάκριση μεταξύ NP και coNP παύει να υφίσταται, άρα μπορούμε να πούμε πως NP = coNP

(c)

Προφανώς ισχύει ότι **NAE3SAT** $\in$ NP. Για να δείξουμε ότι το **NAE3SAT** είναι και NP-δύσκολο, θα αναγάγουμε σε αυτό το **3SAT**, το οποίο γνωρίζουμε ότι είναι NP-πλήρες. Έστω τύπος $\varphi = \bigwedge_{i=1}^{k} C_i$ σε 3 κανονική συζευκτική μορφή. Για κάθε $i \in \{1, ..., k\}$ θεωρούμε μια καινούργια μεταβλητή a_i που δεν εμφανίζεται στον φ, και μια καινούργια μεταβλητή b που δεν εμφανίζεται στον φ και δεν είναι καμία απο τις a_i και θεωρούμε έτσι τον τύπο $\varphi' = \bigwedge_{i=1}^{k} C'_i$. Αν $\phi \in \mathbf{3SAT}$, τότε, υπάρχει αποτίμηση που ικανοποιεί τον φ, άρα ικανοποιεί καθέναν απο τους C_i. Αν $\phi' \in \mathbf{NAE3SAT}$, για κάθε $i \in \{1, ..., k\}$ πρέπει να ικανοποιείται τουλάχιστον ένα απο τα l_{i1}, l_{i2} και l_{i3}, άρα και ο C_i. Συμπεραίνουμε ότι ικανοποιούνται όλοι οι C_i, άρα και ο φ. Οπότε είναι NP-πλήρες.