ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΑΛΓΟΡΙΘΜΟΙ ΚΑΙ ΠΟΛΥΠΛΟΚΟΤΗΤΑ

2η ΣΕΙΡΑ ΓΡΑΠΤΩΝ ΑΣΚΗΣΕΩΝ

Διδάσκοντες:

Άρ. Παγουρτζής
Δ. Φωτάκης
Δ. Σούλιου
Π. Γροντάς

Ειρήνη Δόντη
ΑΜ 03119839

7ο εξάμηνο

Αθήνα 2022

Άσκηση 1: Πολύχρωμος Πεζόδρομος

Για τον υπολογισμό του ελάχιστου πλήθους ημερών που απαιτούνται για να ολοκληρωθεί ο χρωματισμός των πλακών του πεζοδρόμου, μπορούμε να χρησιμοποιήσουμε δυναμικό προγραμματισμό. Υπολογίζουμε το ελάχιστο πλήθος ημερών που απαιτείται για να χρωματίσουμε τις πρώτες i πλάκες (για κάθε i από 1 έως n) και αποθηκεύουμε τα αποτελέσματα στον πίνακα d[]. Στη συνέχεια, μπορούμε να χρησιμοποιήσουμε αυτόν τον πίνακα για να υπολογίσουμε το ελάχιστο πλήθος ημερών που απαιτείται για να χρωματίσουμε όλες τις πλάκες. Θεωρούμε, συγκεκριμένα, τον πίνακα d[i], όπου d[i] θα είναι το ελάχιστο πλήθος ημερών που απαιτείται για να χρωματίσουμε τις πρώτες i πλάκες. Αρχικά, d[1] θα είναι 1, διότι χρειάζεται μία μέρα για να χρωματίσουμε τις πρώτες πλάκες. Στη συνέχεια, για κάθε i από 2 έως n, υπολογίζουμε d[i] ως εξής: d[i] = min(d[j] + cost(j+1, i)) για j < i όπου cost(j+1, i) είναι το συνολικό κόστος χρωματισμού των πλακών από τη θέση j+1 έως τη θέση i, χρησιμοποιώντας το εκάστοτε επιθυμητό χρώμα. Το τελικό αποτέλεσμα θα είναι d[n], δηλαδή το ελάχιστο πλήθος ημερών που απαιτείται για να χρωματίσουμε όλες τις πλάκες από την αρχή ως το τέλος του πεζοδρόμου. Ο αλγόριθμος υπολογίζει το ελάχιστο πλήθος ημερών σε χρόνο $O(n^2)$, όπου n είναι το πλήθος των πλακών στον πεζόδρομο.

Άσκηση 2: String Matching

Για να τροποποιήσουμε τον αλγόριθμο Knuth Morris Pratt χωρίζουμε τη συμβολοσειρά, βάσει του τελεστή μπαλαντέρ, και έπειτα εφαρμόζουμε τον αλγόριθμο Knuth Morris Pratt σε κάθε χωρισμένη υποσυμβολοσειρά. Ελέγχουμε, κάθε φορά, αν κάθε τμήμα αποτελεί υποσυμβολοσειρά. Αν κάποια υποσυμβολοσειρά δεν υπάρχει, ο αλγόριθμος αποτυγχάνει. Επίσης, ελέγχουμε σε γραμμικό χρόνο αν τα τμήματα αυτά είναι συνεχόμενα και συνεπώς η χρονική πολυπλοκότητα παραμένει γραμμική.
Η χρονική πολυπλοκότητα υπολογίζεται ως εξής:
Στην περίπτωση που έχουμε x αστερίσκους, τότε η συμβολοσειρά χωρίζεται, το πολύ, σε x+1 υποσυμβολοσειρές. Οπότε, αναγκαστικά, πρέπει να χρησιμοποιηθούν x δείκτες, όπως γίνεται στην περίπτωση της Διαίρει και βασίλευε στρατηγικής. Οπότε, όλες οι αντιστοιχίες, μπορούν να υπολογιστούν σε χρόνο O(xn + m) και σε χώρο O(m + n), όπου n το μήκος του κειμένου και m το μήκος του μοτίβου.
Στο δοσμένο παράδειγμα, η συμβολοσειρά χωρίζεται σε 2 υποσυμβολοσειρές, οπότε, βάσει των παραπάνω, ο αλγόριθμος εκτελείται σε χρόνο O(m + n), όπου n το μήκος του κειμένου και m το μήκος του μοτίβου.

<u>Άσκηση 3: Συντομότερα Μονοπάτια με Συντομεύσεις Ενδιάμεσων Ακμών</u>

Ο κατευθυνόμενος γράφος δεν έχει αρνητικά βάρη και θεωρούμε ότι m το σύνολο των ακμών και n το σύνολο των κόμβων του γραφήματος:

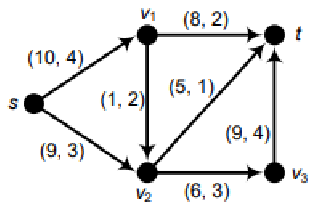
1. Εφαρμόζουμε τον αλγόριθμο Dijkstra στο δοσμένο κατευθυνόμενο γράφημα G με χρήση Fibonacci Heap, ώστε να υπολογίσουμε το συντομότερο μονοπάτι M(v) μεταξύ του s και οποιουδήποτε κόμβου v. Θεωρούμε τον συμπληρωματικό γράφημα G'. Εφαρμόζουμε τον αλγόριθμο Dijkstra στο δοσμένο κατευθυνόμενο γράφημα G' με χρήση Fibonacci Heap, ώστε να υπολογίσουμε το συντομότερο μονοπάτι S(v) μεταξύ του t και οποιουδήποτε κόμβου v. Μηδενίζουμε το μήκος της ακμής (v1, v2) ώστε το άθροισμα M(v1) + S(v2) να είναι το ελάχιστο.
   Οπότε, η χρονική πολυπλοκότητα ώστε να υπολογιστεί το συντομότερο s-t μονοπάτι είναι O(m+nlogn) και η διάσχιση ώστε να βρούμε ποια ακμή θα μηδενίσει το βάρος της είναι της τάξης O(n), γιατί διασχίζουμε στη χειρότερη περίπτωση όλες τις ακμές. Οπότε, η συνολική χρονική πολυπλοκότητα είναι O(m + n + nlogn).

2. Κατασκευάζουμε το γράφημα G' με (k+1)n κορυφές και (k+1)m ακμές, το οποίο προέρχεται από (k+1) αντίγραφα του αρχικού γραφήματος G. Οπότε, στο επεκτεταμένο γράφημα G', για κάθε ακμή από τον κόμβο u στον κόμβο v με βάρος w στο G, υπάρχει ακμή από το (u,i) στο (v,i) με βάρος w, για κάθε i = 1,…,k-1. Στην περίπτωση που στο αρχικό γράφημα G υπάρχει μονοπάτι από τον κόμβο u στον v, τότε στο γράφημα G' υπάρχουν μηδενικού βάρους ακμές από το (u,i) στο (v,i-1), για κάθε i = 1,2,…,k.
   Εφαρμόζουμε τον αλγόριθμο Dijkstra στο δοσμένο κατευθυνόμενο γράφημα με χρήση Fibonacci Heap, ώστε να βρούμε το συντομότερο μονοπάτι από το (s,k) στο (t,0) το οποίο είναι ισοδύναμο με το να βρούμε το συντομότερο μονοπάτι από τον κόμβο s στον κόμβο t στον γράφο G μηδενίζοντας τα βάρη των πολύ k ακμών. Οπότε, η χρονική πολυπλοκότητα ώστε να υπολογιστεί το συντομότερο s-t μονοπάτι είναι O(m+nlogn). Οπότε, η συνολική χρονική πολυπλοκότητα είναι O(m + nlogn).

   Οι παραπάνω αλγόριθμοι είναι ορθοί, στηριζόμενοι στην ορθότητα του αλγορίθμου Dijkstra. Συγκεκριμένα, όταν μία κορυφή u εντάσσεται σε δέντρο συντομότερων μονοπατιών, εφόσον δεν υπάρχουν αρνητικά μήκη, δεν επηρεάζει την απόσταση μίας άλλης κορυφής v από τον αρχικό κόμβο s η οποία είναι μεγαλύτερης τιμής.

Άσκηση 4: Σύντομα μονοπάτια με Επαρκή Μεταφορική Ικανότητα

1. Στο παρακάτω παράδειγμα κατευθυνόμενου γράφου, ο λόγος κόστους προς μεταφορική ικανότητα είναι $\frac{24}{3}$ = 8 και προκύπτει από το μονοπάτι (s, u_2, u_3, t) το οποίο δεν είναι βέλτιστο, αφού το μονοπάτι (s, u_2, t) είναι το βέλτιστο για τον συγκεκριμένο γράφο (το άθροισμα των βαρών είναι 14).



2.

Η αναδρομική σχέση για τον υπολογισμό του βέλτιστου λόγου κόστους προς μεταφορική ικανότητα:

$\text{div}(p, i) = \max\left\{div(p, i-1)\ , \frac{\sum_{e\epsilon p}^{i} c(e)}{min_{e\epsilon p,i}\{b(e)\}}\right\}$ όπου $e\epsilon E$, div(p, 0) = 0 και i=1,…,m.

3.

Θεωρούμε b’, έναν πολύ μεγάλο ακέραιο και δημιουργούμε τον παρακάτω ζητούμενο αλγόριθμο.

i. Αρχικοποιούμε τις μεταβλητές c’ και f με ∞.
ii. Υπολογίζουμε το συντομότερο μονοπάτι p (μέσω του αλγορίθμου Dijkstra με χρήση Fibonacci Heap) της διαδρομής s-t για την οποία η χωρητικότητα γίνεται μέγιστη. Αν δεν υπάρχει το μονοπάτι p ή c(p) ≥ c’, τότε ο αλγόριθμος τερματίζεται.
iii. Αν $\frac{c(p)}{b(p)}$ < f, τότε:
    1. Θέτουμε p’ = p.
    2. Θέτουμε c’ = $\frac{c(p)}{b(p)}$b’ και f = $\frac{c(p)}{b(p)}$
    3. Θέτουμε x = b(p) και συνεχίζουμε στο βήμα

    Αν $\frac{c(p)}{b(p)}$ ≥ f, τότε θέτουμε x = $\frac{c(p)}{f}$ και συνεχίζουμε το επόμενο βήμα.
iv. Διαγράφουμε από το γράφο όλες τις ακμές (i,j) για τις οποίες x ≥ b_{ij}.
v. Επιστροφή στο βήμα ii.

Οπότε, η χρονική πολυπλοκότητα ώστε να υπολογιστεί το συντομότερο s-t μονοπάτι και ο βέλτιστος ζητούμενος λόγος είναι O(m(m+nlogn)).
Ο παραπάνω αλγόριθμος είναι ορθός, στηριζόμενοι στην ορθότητα του αλγορίθμου Dijkstra. Συγκεκριμένα, όταν μία κορυφή u εντάσσεται σε δέντρο συντομότερων μονοπατιών, εφόσον δεν υπάρχουν αρνητικά μήκη, δεν επηρεάζει την απόσταση μίας άλλης κορυφής v από τον αρχικό κόμβο s η οποία είναι μεγαλύτερης τιμής.

Άσκηση 5: Αγορές Προϊόντων από Συγκεκριμένα Καταστήματα

1.

Θεωρούμε διμερές γράφημα G, το οποίο αποτελείται από τις κορυφές όλων των καταστημάτων S και των προϊόντων P. Δημιουργούμε το διμερές συμπλήρωμα του γραφήματος εισόδου G και το ονομάζουμε G'. Οπότε, θα έχουμε ακόμη τις ίδιες κατατμήσεις S και P, αλλά στο νέο γράφημα G' δύο οι κορυφές $s_i \in S$ και $p_i \in P$ συνδέονται αν και μόνο αν δεν συνδέονται στο γράφημα G. Οι πλήρεις διμερείς υπογράφοι του G, αντιστοιχούν σε ανεξάρτητα σύνολα στο G, δηλαδή σύνολα κορυφών που δεν συνδέονται με καμία άμεση ακμή. Οπότε, στο G' αναζητούμε το μέγιστο ανεξάρτητο σύνολο. Αρχικά, αναζητούμε τη μέγιστη διμερή αντιστοίχιση σε G' και έπειτα κατασκευάζουμε ένα πιθανό ελάχιστο κάλυμμα κορυφής του G' με τη βοήθεια του θεωρήματος Kőnig.

Το συμπλήρωμα αυτού του καλύμματος κορυφής είναι ένα μέγιστο ανεξάρτητο σύνολο σε G' δηλαδή το μέγιστο πλήρες διμερές υπογράφημα του G.

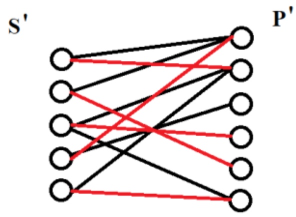
Η χρονική πολυπλοκότητα που χρειάζεται ο παραπάνω αλγόριθμος είναι της τάξης πολυωνυμικού χρόνου O(mn), όπου m ο αριθμός ακμών και n ο αριθμός κορυφών.

2.

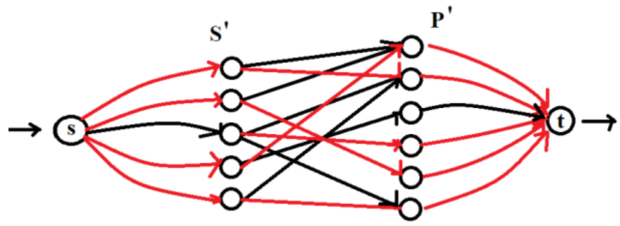
Για το συγκεκριμένο υποερώτημα, θα χρησιμοποιήσουμε τη μέθοδο max Matching σε διμερές γράφημα. Το γράφημα αποτελείται από τις κορυφές όλων των καταστημάτων S και των προϊόντων P. Επίσης, οι κορυφές μεταξύ των καταστημάτων S' και προϊόντων P' συνδέονται με ακμή στην περίπτωση που τα εκάστοτε προϊόντα δεν πωλούνται στα εκάστοτε καταστήματα. Μπορούμε να λύσουμε το πρόβλημα, μειώνοντάς το σε maximum flow πρόβλημα. Έστω G ένας διμερής γράφος με vertex set S' ᵁ P', ώστε κάθε ακμή να συνδέει τις κορυφές στο σύνολο S' με εκείνες στο σύνολο P'. Δημιουργούμε έναν κατευθυνόμενο γράφο G κατευθύνοντας τις ακμές από το σύνολο S' στο P' και προσθέτοντας μία καινούρια πηγή κορυφών s με ακμές σε κάθε κορυφή στο S'. Επίσης, προσθέτουμε μία καινούρια κορυφή-στόχος t με ακμές από κάθε κορυφή στο P', δημιουργώντας, έτσι, τον γράφο G' . Κάθε ταίριασμα M στο γράφο G μπορεί να μετασχηματιστεί σε flow f_M στο G' ως εξής:
Για κάθε ακμή uw στην M, προώθησε μία μονάδα στο μονοπάτι s → u → w → t. Με αυτόν τον τρόπο, το flow ικανοποιεί τους περιορισμούς

χωρητικότητας και συνεπώς η τιμή του flow ταυτίζεται με τον αριθμό των ακμών στην M.
Υπολογίζουμε τα (s,t)-flow f στον γράφο G’, χρησιμοποιώντας τον αλγόριθμο Ford-Fulkerson. Θεωρούμε ότι οι χωρητικότητες των ακμών είναι ακέραιοι αριθμοί, οπότε ο αλγόριθμος Ford-Fulkerson καταγράφει έναν ακέραιο (που υποδεικνύει το flow) σε κάθε ακμή. Το μέγεθος ταιριάσματος είναι ακριβώς |f| και επιπλέον το μέγεθος του μέγιστου ταιριάσματος στο G ταυτίζεται με την τιμή του μέγιστου ταιριάσματος στο G’. Οπότε, η χρονική πολυπλοκότητα του αλγορίθμου είναι της τάξης O(VE) με χρήση του αλγορίθμου Ford-Fulkerson, όπου V οι κορυφές και E οι ακμές του γράφου.
Για παράδειγμα, θεωρούμε δύο απεικονίσεις των γράφων G και G’:

Ο γράφος G



Ο γράφος G’



Οι παραπάνω αλγόριθμοι είναι ορθοί, καθώς περικλείουν ένα ταίριασμα M στο G το οποίο είναι αρχικά είναι κενό. Οι ακμές του M αντιστοιχίζονται σε ακμές στο G’ που έχουν flow. Ταιριάζουμε μία κορυφή του γράφου G αν είναι το άκρο κάποιας ακμής στο M. Σε κάθε επανάληψη, αναζητείται για εναλλακτικό μονοπάτι στον γράφο G, το οποίο αποτελείται από μία αταίριαστη κορυφή στο L σε μία αταίριαστη κορυφή στο R. Αν βρεθεί ένα augmenting μονοπάτι P, αυξάνουμε τον αριθμό των ακμών στο M κατά 1 και συνεχίζουμε με την επόμενη επανάληψη. Αν δεν υπάρχει κάποιο εναλλακτικό μονοπάτι, τότε το M είναι το μέγιστο ταίριασμα και ο αλγόριθμος τελειώνει.

Άσκηση 6: Ενοικίαση Αυτοκινήτων

Η εταιρία διαθέτει k αυτοκίνητα προς ενοικίαση και υπάρχουν n προσφορές από τους πελάτες. Κάθε προσφορά (s_i, t_i, p_i) αναφέρει τη χρονική στιγμή δέσμευσης του αυτοκινήτου s_i , τη χρονική στιγμή αποδέσμευσης του αυτοκινήτου t_i και το αντίτιμο p_i που προσφέρει. Αν η εταιρία αποδεχθεί την προσφορά, δεσμεύει ένα από τα αυτοκίνητα προς ενοικίαση για το χρονικό διάστημα [s_i, t_i) και εισπράττει το αντίτιμο p_i. Θεωρούμε γράφο με k κορυφές στις οποίες θα ενώνονται n ακμές-προσφορές με βάρη την εκάστοτε προσφορά (s_i, t_i, p_i) των πελατών. Θα εφαρμόσουμε αναγωγή μετασχηματισμού σε min-cost flow με τη βοήθεια του αλγορίθμου Klein. Δημιουργούμε το υπολειμματικό δίκτυο G_f του γράφου, αφαιρώντας τη χωρητικότητα και τη ροή των βαρών (s_i, t_i) ως ότου να μην εμφανιστεί κύκλος αρνητικού κόστους. Βρίσκουμε μία πιθανή ροή f και όσο υπάρχει κύκλος αρνητικού κόστους X στο υπολειμματικό δίκτυο G_f , θεωρούμε το ελάχιστο του βάρους των

ακμών στο X ως μονάδες ροής γύρω από το X. Με αυτό τον τρόπο, εξασφαλίζουμε ότι η ροή είναι ελαχίστου κόστους. Εφόσον, οι χωρητικότητες και η ροή των βαρών είναι φυσικοί αριθμοί, τότε η χρονική πολυπλοκότητα του αλγορίθμου είναι της τάξης $O(kn^2)$, εφόσον γίνονται επαναλήψεις πολλαπλάσιες των ακμών του γράφου.
Η ορθότητα του αλγορίθμου, βασίζεται στην ορθότητα του αλγορίθμου Klein, ο οποίος αρχικοποιεί την f ως μηδέν και στη συνέχεια αυξάνει επανειλημμένα το f κατά μήκος ενός αυθαίρετου αρνητικού υπολειπόμενου κύκλου, έως ότου δεν υπάρχουν άλλοι αρνητικοί υπολειπόμενοι κύκλοι.