Τουρνή Ισιδώρα-Χαρά
ΑΜ 03111403

Αλγόριθμοι και Πολυπλοκότητα
4η Σειρά Γραπτών Ασκήσεων
Ημ/νια παράδοσης 20/2/2015

Άσκηση 1 - Επιβεβαίωση και Αναπροσαρμογή Συντομότερων Μονοπατιών.

- ισχυρά συνεκτικό $G(V, E, w)$, n κορυφές, m ακμές, (ενδεχομένως αρνητικά μήκη w στις ακμές). $d(u, v)$ = απόσταση των u, v στο G.

a) n αριθμοί $\delta_1, \delta_2, \ldots \delta_n$, όπου δ_k = η απόσταση $v_1 - v_k$ στο G. (υποθετικά).
Θα τροποποιήσουμε τον αλγόριθμο του BFS καταλλήλως, ώστε να πραγματοποιεί το απαιτούμενο έλεγχο και εν τέλει να μας επιστρέφει το shortest paths tree.
Ως κορυφή έναρξης s θεωρούμε την $V1$ και $p[vi]$ = NIL για όλες τις κορυφές του G. Εξετάζουμε την κορυφή v_i, η οποία έστω ότι γειτονεύει με την κορυφή v_j, και ελέγχουμε το άθροισμα $\delta_i + w_{ij}$. Εάν $\delta_i + w_{ij} < \delta_j$ τότε οι δοθείσες αποστάσεις είναι λανθασμένες. Διαφορετικά, αν $\delta_i + w_{ij} = \delta_j$, η απόσταση για την j είναι σωστή και προχωράμε στον αλγόριθμο του BFS, θεωρώντας και $p[j]=i$. Στο τέλος του αλγορίθμου, αν ο πίνακας p περιέχει σε κάποια θέση (εκτός της 1ης) την τιμή NIL, τότε οι αποστάσεις $\delta_1, \ldots \delta_n$ είναι λανθασμένες. Αλλιώς, είναι σωστές και το shortest paths tree δίνεται από τον πίνακα p. Η πολυπλοκότητα του αλγορίθμου είναι αυτή του BFS, δηλαδή $O(n+m)$.

b) $d(u_i, u_j)$, $(u_i, u_j) \in V \times V$ μήκος της $e=(x,y) \rightarrow w'(x,y) < w(x,y)$
Η απόσταση $d(u_i, u_j)$ μεταξύ των u_i, u_j θα αλλάξει (θα μειωθεί) μόνο εάν το μήκος του μονοπατιού που περνά από την e (με w') είναι μικρότερο από το $d(u_i, u_j)$. Το ζητούμενο αυτό μήκος δίνεται από το άθροισμα $d(u_i, x) + w'(x, y) + d(y, u_j)$, και μπορεί να υπολογισθεί για όλα τα ζεύγη των κορυφών $e=(x, y)$ σε χρόνο $O(n^2)$ (θεωρώντας ότι έχουμε $O(n^2)$ κορυφές)

γ) $e=(x,y) \rightarrow w'(x,y) > w(x,y)$
Με την αύξηση του μήκους μιας ακμής, ο έλεγχος της απόστασης $d(u_i, x) + w'(x,y) + d(y, v_j)$ δεν αρκεί, καθώς θα χρειαστεί πιθανόν να εξετάσουμε και όλα τα υπόλοιπα μονοπάτια μεταξύ των u_i, u_j, που δε χρησιμοποιούν την e. Εάν $d(u_i, x) + w'(x, y) + d(y, v_j) = d(u_i, u_j)$ δεν έχουμε καμία αλλαγή στην απόσταση d. Εάν ισχύει $d(u_i, x) + w'(x, y) + d(y, v_j) > d(u_i, u_j)$ τότε, είναι αναγκαίο να δούμε και όλα τα υπόλοιπα μονοπάτια μεταξύ των κορυφών v_i, u_j, καθώς μπορεί κάποιο να έχει μικρότερο μήκος, οπότε και να αναπροσαρμόσουμε την απόσταση.

Άσκηση (2) – Σύστημα Ανισοτήτων και Μετατροπή Σταθερών Όρων

- $x_1, \dots x_n$ ακέραιες μεταβλητές
- σύστημα S αποτελούμενο από m ανισότητες της μορφής $x_i - x_j \le b_{ij}$, για
  κάποια $1 \le i,j \le n$ όπου τα b_{ij} ακέραιοι

S ικανοποιήσιμο αν υπάρχουν ακέραιες τιμές για τις $x_1, \dots x_n$ που
ικανοποιούν όλες τις ανισότητες.

α) Θα μοντελοποιήσουμε τις μεταβλητές και το σύστημα S των ανισο-
τήτων σε ένα γράφο G(V,E), θεωρώντας μία κορυφή για κάθε μετα-
βλητή x_i, την v_i, και ενώνοντας τις v_i, v_j, για τις οποίες ισχύει $x_i - x_j \le b_{ij}$
με ακμή μήκους b_{ij}. Το κριτήριό μας συνοψίζεται στο ότι το σύστημα
S είναι ικανοποιήσιμο ⟺ ο γράφος G δεν περιέχει κύκλο αρνητικού
μήκους.

Στην περίπτωση που ο G έχει κύκλο αρνητικού μήκους, τον
$v_1, v_2, \dots v_\ell, v_1$, με μήκος $\ell<0$, έχοντας τις ανισότητες $v_1 - v_2 \le b_{12}$,
$\dots v_\ell - v_1 \le b_{\ell 1}$, τις προσθέτουμε κατά μέλη, οπότε το πρώτο μέλος
μηδενίζεται και στο δεύτερο θα έχουμε μια αρνητική ποσότητα (κύκλος
αρνητικός). Δηλαδή προκύπτει $0 \le \ell < 0$, κάτι που είναι άτοπο,
και άρα το σύστημα δε μπορεί να είναι ικανοποιήσιμο.

Αν το γράφημα δεν έχει αρνητικό κύκλο, τότε μπορούμε να εφαρ-
μόσουμε τον αλγόριθμο Bellman-Ford για την εύρεση του shortest
paths tree και τον εντοπισμό αρνητικών κύκλων. Ως αρχική κορυφή
s για τον αλγόριθμο θεωρούμε μια εξωτερική κορυφή s, την οποία
συνδέουμε με τις υπόλοιπες με ακμές μηδενικού μήκους. Βρίσκουμε το
shortest paths tree, και σε κάθε μεταβλητή x_i μεταφέρουμε την τιμή της
απόστασης από την s. Όλες οι ανισότητες θα ικανοποιούνται, οι μεν
που ανήκουν στο shortest paths tree ως ισότητες, ενώ όσες δεν ανήκουν
ικανοποιούνται εξαιτίας της τριγωνικής ανισότητας στις αποστάσεις
των shortest paths, και επομένως το S είναι ικανοποιήσιμο.

Άρα ο αλγόριθμος για να αποφασίσουμε αν το S είναι ικανοποι-
ήσιμο προκύπτει με εφαρμογή του Bellman-Ford: σε κάθε βήμα
ελέγχουμε εάν ελαττώνεται η απόσταση προς κάποια εκ των n κορυ-
φών, ελέγχοντας όλες τις ακμές. Μετά από n-1 βήματα, εάν υπάρχει
ακόμη ακμή που ελαττώνει την απόσταση προς κάποια κορυφή, τότε
υπάρχει κύκλος αρνητικού μήκους, και άρα το S δεν είναι ικανοποιήσιμο.
Η πολυπλοκότητα του αλγορίθμου είναι $O(n \cdot m)$ → κάθε ακμή θα εξετασθεί
n-1 φορές.

Αν το S είναι ικανοποιήσιμο, τότε, με τη βοήθεια του πίνακα p (προγό-
νων) του αλγορίθμου κατασκευάζουμε το shortest paths tree από την
κορυφή v_1 (μεταβλητή x_1). Οι αποδεκτές τιμές που ανατίθενται στις με-
ταβλητές x_i θα είναι, όπως προαναφέρθηκε, οι αποστάσεις των v_i από
την κορυφή v_1 (με $x_1 = 0$), οπότε και το S είναι ικανοποιήσιμο. Και εδώ
η πολυπλοκότητα είναι $O(n \cdot m)$.

β) S', $x_i - x_j \le b_{ij}'$, με $b_{ij}' \ge 0$, $1 \le j, i \le n$ (3)

- S', S ισοδύναμα όταν S' ικανοποιήσιμο $\Leftrightarrow$ S ικανοποιήσιμο.
- τροποίηση σταθερών όρων του S μέσω της Change(S,i,k), που προσθέτει τον αριθμό k σε όλα τα b_{ij}, $1 \le j \le n$ και αφαιρεί τον k από όλα τα b_{li}, $1 \le l \le n$.
- ; με διαδοχ. κλήσεις της change : $S \to S'$

Θα χρησιμοποιήσουμε μια παραλλαγή του αλγορίθμου του Johnson, ώστε να μετασχηματίσουμε το γράφο που δημιουργήσαμε στο ερώτημα α, με τα στοιχεία x_i του S, στον G' που θα αντιστοιχεί στο σύστημα S', με $b_{ij}' \ge 0$.

Αρχικά δημιουργούμε μία νέα κορυφή q, η οποία προστίθεται στον G και συνδέεται με ακμές μηδενικού μήκους με τις υπόλοιπες κορυφές. Χρησιμοποιώντας τον αλγόριθμο Bellman-Ford, και με εναρκτήρια κορυφή την q, βρίσκουμε για καθεμιά από τις υπόλοιπες κορυφές v το μήκος του συντομότερου μονοπατιού από την q στη v, και το αποθηκεύουμε στον πίνακα $h[v]$. (σε περίπτωση που βρεθεί κύκλος αρνητικού μήκους, ο αλγόριθμος τερματίζει)

Χρησιμοποιώντας τώρα τη συνάρτηση change, τροποποιούμε τα μήκη των ακμών του γραφήματος G' ως εξής: καλούμε την change για κάθε κορυφή i, ως change$(S, i, h[v_i])$, $1 \le i \le n$, και εν τέλει παίρνουμε το γράφο G' που απεικονίζει το σύστημα S', με $x_i - x_j \le b_{ij}'$, $b_{ij}' \ge 0$, $1 \le j, i \le n$.

Ο παραπάνω αλγόριθμος μετατροπής $S \to S'$ έχει πολυπλοκότητα $O(n \cdot m)$. Η ορθότητά του προκύπτει από την ορθότητα του αλγορίθμου Johnson, ο οποίος υπολογίζει τα νέα μη αρνητικά μήκη. Στον πίνακα h δίνονται οι αποστάσεις κάθε κορυφής από την s. Για κάθε κορυφή v σε κάθε εξερχόμενη ακμή προστίθεται η ποσότητα $h[v]$ και σε κάθε εισερχόμενη σε αυτή ακμή αφαιρείται το $h[v]$. Επομένως το τελικό μήκος κάθε ακμής, άρα κάθε μονοπατιού $p = (x = v_0, v_1, \ldots v_k = y)$ θα είναι

$$b_{0k}' = \sum_{i=0}^{k-1} b_{i,i+1}' = \sum_{i=0}^{k-1} [b_{i,i+1} + h[v_i] - h[v_{i+1}]] =$$

$$= \sum_{i=0}^{k-1} b_{i,i+1} + h[v_0] - h[v_k] = b_{0k} + h[x] - h[y]$$

Όπως και στο ερώτημα α, μπορούμε να δείξουμε πότε το S' είναι ικανοποιήσιμο, και να βρούμε το shortest paths tree για το G'.

Μπορούμε να παρατηρήσουμε ότι, εάν S' ικανοποιήσιμο, και αναθέσαμε τιμές στις μεταβλητές x_i τις αποστάσεις της κορυφής v_i από την v_1 (με $x_1 = 0$), τότε μπορούμε εύκολα να βρούμε τις αποδεκτές τιμές για τις μεταβλητές του S, προσθέτονται σε κάθε μεταβλητή το $h[vi]$ της αντίστοιχης κορυφής vi, οπότε και το S θα είναι ικανοποιήσιμο με τις παραπάνω τιμές. Δηλαδή (4)

$$x_i\ (\in S) = x_i\ (\in S') + h[vi]$$

Άσκηση 4 - Επιβεβαίωση και αναπροσαρμογή Μέγιστης Ροής

- κατευθυνόμενο s-t δίκτυο $G(V, E, c)$ με n κορυφές, m ακμές, (θετικές) ακέραιες χωρητικότητες c στις ακμές

α) ροή f: υποθετικά μέγιστη ροή στο G.

Για το ζητούμενο έλεγχο, θα χρησιμοποιήσουμε τον αλγόριθμο Ford-Fulkerson, ως βασική ιδέα, για να αποφασίσουμε αν η δοθείσα ροή είναι η μέγιστη. Δημιουργούμε το υπολειμματικό δίκτυο $G_f(V, E_f, r_f)$ και με BFS αναζητούμε αν υπάρχει επαυξητικό μονοπάτι (= απλό μονοπάτι s-t στο δίκτυο G_f), που αποτελεί και τη συνθήκη ελέγχου για τον τερματισμό του Ford-Fulkerson. Εάν δεν υπάρχει, τότε η f που δόθηκε είναι μέγιστη.

β) f μέγιστη ροή στο G

- η πραγματική χωρητικότητα μιας ακμής e είναι μικρότερη κατά k μονάδες, $1 \leq k \leq c_e$, από τη c_e που είχαμε θεωρήσει αρχικά. Θα χρησιμοποιήσουμε ξανά τον αλγόριθμο Ford-Fulkerson, ο οποίος όμως αρχικά απαιτεί να έχουμε μια ροή f' με σκοπό την κατασκευή του υπολειμματικού δικτύου $G_{f'}$. Επομένως θα πρέπει να υπολογίσουμε αυτή την εφικτή ροή.

Αν η νέα χωρητικότητα $c_e' \geq f_e$, τότε δε θα χρειαστεί κάποια αλλαγή και παίρνουμε ως ροή αυτή που έχουμε ήδη, την f. Διαφορετικά θα υπολογίσουμε την f' ως εξής:

- Θέτουμε $f_e' = c_e'$, για την ακμή $e = (x, y)$. Θα εκτελέσουμε δύο BFS: Στο πρώτο θα διασχίσουμε το υπογράφημα από το x στο s και στο άλλο από το y στο t. Θέλουμε, ξεκινώντας με τη μείωση από το x (προχωρώντας προς το s) και μετά όμοια από το y (προς το t) να βρούμε μια f'. Θεωρούμε κορυφή v_i, στην οποία φθάνει νέα ροή f_i', και θέλουμε να μειώσουμε τη ροή στις προσπίπτουσες ακμές κατά k. Μειώνουμε τη ροή προς μια γειτονική v_j μέχρι, είτε η ροή της ακμής να μηδενιστεί, είτε να έχουμε φθάσει στις k μονάδες ροής. Επαναλαμβάνουμε την παραπάνω διαδικασία για όλες τις υπόλοιπες γειτονικές κορυφές της v_i, μέχρι να μειώσουμε τη συνολική ροή κατά k μονάδες. Με την ολοκλήρωση αυτής, έχουμε βρει την εφικτή ροή f', οπότε και εφαρμόζουμε τον αλγόριθμο Ford-Fulkerson.

Η πολυπλοκότητα του αλγορίθμου προκύπτει από αυτή του Ford-Fulkerson, και το γεγονός ότι θα εκτελεσθούν το πολύ k επαναλήψεις για την εύρεση της f' και άρα του υπολειμματικού δικτύου $G_{f'}$, και άρα θα είναι $O(k(V+E))$.

Άσκηση 5 - Αναγωγές και NP-πληρότητα

• **Συνδετικό Δέντρο Περιορισμένο ως προς το Μέγιστο Βαθμό**

- Αρχικά θα δείξουμε ότι το πρόβλημα είναι στο NP. Υποθέτουμε ότι έχουμε ένα δέντρο T στο γράφημα G. Μπορούμε να ελέγξουμε ότι κάθε κορυφή u στο T έχει βαθμό το πολύ 2 σε γραμμικό χρόνο, εξετάζοντας κάθε ακμή $e=(u,v)$. Επιπλέον, μπορούμε να δείξουμε ότι μπορούμε να φθάσουμε κάθε κορυφή από κάποια εναρκτήρια κορυφή s και ότι το γράφημα δεν περιέχει κύκλους με DFS, επίσης σε γραμμικό χρόνο. Αυτά αρκούν για να δείξουμε ότι το T είναι ένα spanning tree με κορυφές βαθμού το πολύ 2 του γραφήματος G και άρα ανήκει στο NP.
- Εν συνεχεία θα αποδείξουμε ότι είναι NP πλήρες, με αναγωγή του προβλήματος του μονοπατιού Hamilton, που είναι NP-πλήρες. Θεωρούμε ένα μονοπάτι Hamilton για ένα γράφο G. Θεωρούμε $k=2$
  - Αν υπάρχει spanning tree με βαθμό κορυφών το πολύ 2 στο G, τότε θα υπάρχει μονοπάτι Hamilton στο G, που είναι ακριβώς αυτό το δέντρο (μονοπάτι Hamilton: απλό μονοπάτι που διέρχεται από όλες τις κορυφές του G).
  - Αν δεν υπάρχει το ζητούμενο spanning tree στο G, τότε δεν υπάρχει μονοπάτι Hamilton στο G. Υποθέτουμε το αντίθετο, ότι δεν υπάρχει τέτοιο spanning tree στο G, αλλά υπάρχει μονοπάτι Hamilton P, το οποίο περιλαμβάνει όλες τις κορυφές του G και δεν περιέχει κύκλους. Επομένως, ο βαθμός κάθε κορυφής στο P είναι το πολύ 2. Όμως αυτό είναι ένα spanning tree με βαθμό κάθε κορυφής το πολύ 2, άρα ερχόμαστε σε άτοπο.

Δείξαμε επομένως ότι το δέντρο περιορισμένο ως προς το μέγιστο βαθμό κάθε κορυφής.

• Διαχωρισμός σε Κλάσεις (Clustering)

Θα ανάγουμε το πρόβλημα του k-χρωματισμού ενός γράφου στο πρόβλημα Clustering. Στο πρώτο, με δεδομένο ένα γράφημα $G=(V,E)$ εξετάζουμε εάν οι κορυφές μπορούν να χρωματισθούν με το πολύ k χρώματα, ώστε οι δύο κορυφές που αποτελούν τα άκρα κάθε ακμής να μην έχουν το ίδιο χρώμα, δηλαδή $col(i) \neq col(j)$ για κάθε $(i,j) \in E$.
Αντιστοιχούμε τους κόμβους $v \in V$ στα σημεία p_i. Θεωρούμε τη συνάρτηση της απόστασης $d(p_i, p_j)$ έτσι ώστε

$$d(p_i,p_j) = \begin{cases} cB, & \text{if } (u,v) \in E \\ B/c, & \text{if } (u,v) \notin E \\ 0, & \text{if } u=v \end{cases}$$

για κάποια σταθερά $c>1$ και κάθε φυσικό $B>0$

- Αν υπάρχει λύση S στο πρόβλημα Clustering, τότε αυτή είναι και λύση στο πρόβλημα του k-χρωματισμού. Αφού δεν υπάρχει υποσύνολο $P \in S$ που να περιέχει σημεία $p_i, p_j \in P$ τέτοια ώστε $d(p_i,p_j) > B$, γνωρίζουμε ότι $(i,j) \notin E$. Επομένως τα i,j μπορούν να έχουν το ίδιο χρώμα. Χρωματίζουμε τις κορυφές με τέτοιο τρόπο έτσι ώστε για το $p_i \in P_\ell$ να ισχύει $col(v) = C(P_\ell)$, όπου $C(\cdot)$ μοναδικό χρώμα για κάθε κλάση $P_\ell \in S$ έτσι ώστε $C(P_i) \neq C(P_j)$ αν $i \neq j$.

- Αντίστροφα, αν υπάρχει λύση στο πρόβλημα του k-χρωματισμού ενός γράφου, τότε υπάρχει λύση στο πρόβλημα του Clustering. Αφού $col(i) \neq col(j)$ για όλους τους κόμβους i,j έτσι ώστε $(i,j) \in E$, γνωρίζουμε ότι $d(p_i,p_j) > B$. Επομένως τα σημεία p_i και p_j δε μπορούν να ανήκουν στην ίδια κλάση $P \in S$. Διαχωρίζουμε τα σημεία έτσι ώστε για κάθε χρώμα $c_1, \dots c_k$ να ορίσουμε κλάσεις $P_1, \dots P_k$ και θεωρούμε ότι $p_i \in P_\ell$ αν $col(i) = c_\ell$.

Συντομότερο Μονοπάτι με Περιορισμούς (Constraint Shortest Path)

Θα ανάγουμε το πρόβλημα Knapsack στο πρόβλημα του Constraint Shortest Path.

Το γενικό στιγμιότυπο του knapsack είναι το ακόλουθο: έχουμε n στοιχεία, το καθένα με βάρος w_i και κέρδος p_i.

Το ερώτημα που τίθεται είναι εάν υπάρχει υποσύνολό τους με συνολικό βάρος μικρότερο ή ίσο του B και συνολικό κέρδος μεγαλύτερο ή ίσο του P. Θα μετασχηματίσουμε το παραπάνω σε ένα ειδικό στιγμιότυπο του προβλήματος που δόθηκε.

Δημιουργούμε νέο γράφημα G', κατευθυνόμενο, με x_0 αρχική κορυφή και x_i, $i=1$ έως n οι υπόλοιπες κορυφές, οι οποίες αντιστοιχούν στα στοιχεία του σακιδίου.

Διαπιστώνουμε ότι υπάρχουν δύο κατευθυνόμενες ακμές από κάθε x_{i-1} στο x_i. Η μία αντιστοιχεί στην επιλογή του στοιχείου i για το σακίδιο με $w(e)=w_i$ και $c(e)=P_{sum}-p_i$. Η δεύτερη ακμή αντιστοιχεί στη μη επιλογή του στοιχείου i για το σακίδιο και έχει $w(e)=0$ και $c(e)=P_{sum}$ = συνολικό κέρδος. Θέτουμε επιπλέον $s=x_0$, $t=x_n$, $W=B$, $C=n\cdot P_{sum}-P$.

Διακρίνουμε ότι η εύρεση ενός αποδεκτού σακιδίου ισοδυναμεί με την εύρεση ενός μονοπατιού από το s στο t που να πληροί τους παραπάνω περιορισμούς.