Αλγόριθμοι και Πολυπλοκότητα

4η σειρά γραπτών ασκήσεων

Σχέδιο Λύσεων

CoReLab

ΣΗΜΜΥ Ε.Μ.Π.

12 Φεβρουαρίου 2020

Άσκηση 1α: Επιβεβαίωση Συντομότερων μονοπατιών

Ο αλγόριθμος είναι μια απλή παραλλαγή του BFS

- Ξεκινάμε το BFS από την κορυφή v_1 με $p(v_1) = NULL$
- Έστω ότι είμαστε στην κορυφή v_i και έστω v_j γειτονάς του.
- Αν $\delta_i + w_{ij} < \delta_j$ οι αποστάσεις είναι λάθος. Αν $\delta_i + w_{ij} = \delta_j$, σημειώνουμε την κορυφή και ανανεώνουμε τον πίνακα προγόνων $p[j] = i$. Αλλιώς, προχωράμε.
- Αν ολοκληρωθεί η διαδικασία ελέγχουμε τον πίνακα προγόνων εάν έχει σε κάποια θέση εκτός την πρώτη $NULL$. Αν έχει, τότε απορρίπτουμε τις δοθείσες αποστάσεις. Αλλιώς, επιστρέφουμε τον πίνακα προγόνων που αποτελεί το ΔΣΜ.
- Η πολυπλοκότητα είναι $O(n + m)$.

Άσκηση 1β και 1γ: Αναπροσαρμογή Συντομότερων Μονοπατιών

- Μειώνεται το μήκος της ακμής $e = (x, y)$ σε $w'(x, y) < w(x, y)$. Άρα για κάθε ζεύγος κορυφών v_i, v_j υπάρχει περίπτωση να μειωθεί η απόσταση μεταξύ τους μόνο εάν το μήκος του μονοπατιού που περνά από την e είναι μικρότερο από το δοθέν $d(v_i, v_j)$.
- Το μήκος αυτό είναι $d(v_i, x) + w'(x, y) + d(y, v_j)$ και υπολογίζεται προφανώς σε ένα υπολογιστικό βήμα. Έτσι εφόσον έχουμε $O(n^2)$ ζεύγη, η πολυπλοκότητα του αλγορίθμου είναι $O(n^2)$.

Άσκηση 1β και 1γ: Αναπροσαρμογή Συντομότερων Μονοπατιών

- Αν αυξηθεί το μήκος της ακμής $e = (x, y)$ σε $w'(x, y) > w(x, y)$, δεν αρκεί στη γενική περίπτωση να συγκρίνουμε τη δοθείσα απόσταση $d(v_i, v_j)$ με το μήκος **ενός μόνο** μονοπατιού.
  Αν ισχύει πως $d(v_i, x) + w'(x, y) + d(y, v_j) = d(v_i, v_j)$ δεν χρειάζεται να αλλάξουμε τίποτα.
  Αν όμως $d(v_i, x) + w'(x, y) + d(y, v_j) > d(v_i, v_j)$ τότε θα πρέπει αναγκαστικά να συγκρίνουμε το $d(v_i, v_j)$ με το μήκος όλων των υπολοίπων μονοπατιών μεταξύ v_i και v_j.

Κριτήριο

Φτιάχνουμε ένα γράφημα $G(V, E)$ όπου:

- Για κάθε μεταβλητή x_i βάζουμε μία κορυφή v_i
- Για κάθε περιορισμό $x_i - x_j \leq b_{i,j}$ βάζουμε μια κατευθυνόμενη ακμή βάρους $w(v_j, v_i) = b_{i,j}$ από την v_j στην v_i.

Το σύστημα ανισοτήτων S είναι ικανοποιήσιμο εάν και μόνο εάν το γράφημα G δεν έχει αρνητικό κύκλο

Ορθότητα Κριτηρίου (ικανή συνθήκη)
Αν το γράφημα δεν έχει αρνητικούς κύκλους, υπάρχει λύση για το σύστημα:

- Συνδέουμε μια νέα κορυφή v_0 με όλες τις κορυφές το γραφήματος, με ακμές βάρους 0.
- Εφαρμόζουμε Bellman-Ford ξεκινώντας από την v_0 και αναθέτουμε σε κάθε μεταβλητή x_i τιμή ίση με την απόστασή της κορυφής της από την v_0, έστω $dist(v_0, v_i)$ (αφού δεν έχουμε αρνητικούς κύκλους, μπορούμε να θέσουμε για κάθε κόμβο αυτήν την απόσταση).

Έτσι, θα ισχύει η τριγωνική ανισότητα:
$dist(v_0, v_i) \leq ds(v_0, v_j) + w(v_j, v_i)$ όπου $w(v_j, v_i) = b_{i,j}$,
$dist(v_0, v_i) = x_i$, $dist(v_0, v_j) = x_j$.

Άρα, τελικά: $x_i - x_j \leq b_{i,j}$, που είναι το ζητούμενο.

Ορθότητα Κριτηρίου (αναγαία συνθήκη)
Έστω ότι το γράφημα έχει αρνητικό κύκλο, τον $v'_1, v'_2, ..., v'_k, v'_1$ με μήκος $L < 0$. Προσθέτοντας τις αντίστοιχες ανισότητες κατά μέλη, έχουμε:

- $x_2 - x_1 + x_3 - x_2 + ... + x_k - x_{k-1} + x_1 - x_k \leq$ $b_{2,1} + b_{3,2} + ... + b_{k,k-1} + b_{1,k} = L$
- Όμως, το πρώτο μέλος της παραπάνω ανισότας είναι 0.
- Άρα, από υπόθεση: $0 \leq L < 0$, Άτοπο.

Άρα για οποιαδήποτε ανάθεση τιμών στις μεταβλητές, το σύστημα δεν μπορεί να είναι ικανοποιήσιμο.

Αποδοτικός Αλγόριθμος

Όπως είδαμε στην απόδειξη ορθότητας του κριτηρίου, κατασκευάζουμε τον γράφο, προσθέτουμε τον κόμβο v_0 και τρέχουμε Bellman-Ford για να ανιχνεύσουμε την ύπαρξη αρνητικών κύκλων. Χαρακτηρίζουμε το σύστημα μη ικανοποιήσιμο, σε περίπτωση ύπαρξης αρνητικού κύκλου.

Έτσι, η πολυπλοκότητα είναι αυτή του Bellman-Ford: $\mathcal{O}(n \times m)$.

Άσκηση 2β

- Στην περίπτωση που το S είναι ικανοποιήσιμο, ο Bellman-Ford στον γράφο που προαναφέρεται επιστρέφει μια εφικτή λύση, (θέτοντας $x_i = dist(v_0, v_i)$), με πολυπλοκότητα $O(n * m)$.
- Για να βρούμε το ελάχιστο 'υπο-σύστημα' εξισώσεων που δεν ικανοποιείται, αρκεί να βρούμε τον ελάχιστο (σε μήκος) αρνητικό κύκλο του γραφήματος.

Άσκηση 2β

Όπως ξέρουμε, μπορούμε να χρησιμοποιήσουμε τον Bellman-Ford για να ανιχνεύσουμε αρνητικούς κύκλους που περέχονται σε 'μονοπάτια' που καταλήγουν στην αρχική κορυφή του αλγορίθμου, v.

Πιο προσεκτικά, αν δούμε τη λειτουργία του BF (στην k-οστή επανάληψη, έχουμε τα ελάχιστα μονοπάτια από τον v με μήκος μέχρι k), βλέπουμε ότι αν ξεκινήσουμε από κάποια κορυφή που ανήκει σε έναν αρνητικό κύκλο μήκους k, μετά από k επαναλήψεις η απόσταση της από τον εαυτό της (αρχικά 0) θα γίνει αρνητική.

Άσκηση 2β

Έτσι, μπορούμε να κατασκευάσουμε τον εξής αλγόριθμο:

Αλγόριθμος

- Αρχικά, θέσε $k = n - 1$
- Για κάθε κόμβο v_i τρέξε τις πρώτες k επαναλήψεις του BF, ή μέχρις ότου η απόσταση του v_i από τον εαυτό του γίνει αρνητική.
  - Αν ικανοποιηθεί η παραπάνω συνθήκη, αποθήκευσε το 'μονοπάτι' (κύκλο) στο οποίο έχει τοποθετήσει ο BF μέχρι στιγμήστην v_i ως P_{nmin} και θέσε ως k τον αριθμό της επανάληψης στον οποίο σταμάτησε ο αλγόριθμος.
- Επέστρεψε τα P_{nmin} και k.

Στη χειρότερη περίπτωση θα τρέξουμε n πλήρεις BF, οπότε:

Πολυπλοκότητα

$$\mathcal{O}(n^2 \times m)$$

Άσκηση 2γ

Για το ερώτημα, θα ονομάζουμε τα βάρη $w_{i,j}$ που προσθέτονται από την εκφώνηση ως *κόστη* (διαφορετικά από τα βάρη $w(i,j) = b_{j,i}$ που έχουμε ήδη ορίσει για τις ακμές).

Ας θυμηθούμε λίγο τον βασικό κύκλο επανάληψης του ΔΠ του Bellman-Ford (χωρίς αρχικοποιήσεις κτλ):

Bellamn-Ford loop

- Επανέλαβε $n-1$ φορές:
  - Για κάθε ακμή (u,v), βάρους $w(u,v)$ $(= b_{v,u})$:
    Αν $dist(v) + b_{v,u} < dist(v)$
    θέσε $dist(v) = dist(v) + b_{v,u}$, $par(v) = u$

Όπου αρχικά, αν τρέχουμε BF στον κόμβο v_0, $dist(v_i) = \infty$ $\forall v_i \in \{1, 2, .., n\}$, $dist(v_0) = 0$ και $par(v_i) = null$ $\forall i \in \{0, 1, ..., n\}$.

Άσκηση 2γ

Θα αλλάξουμε τον αλγόριθμο, έτσι ώστε να μας βρίσκει το ελάχιστο μονοπάτι, κόστους το πολύ C. Τώρα, θα έχουμε ένα memoization array $dist$ δυο διαστάσεων, όπου $dist(v, c)$ η ελάχιστη απόσταση του v με κόστος το πολύ c.
Έτσι, θα προκύψει η αναδρομική σχέση:

$$dist(v, c) = \min_{(v,u)}\{dist(u, c - w_{v,u}) + b_{u,v}\}$$

Συνεπώς τρέχουμε το βασικό *loop* του BF για i από 1 μέχρι C, χρησιμοποιώντας αυτήν την σχέση, φροντίζοντας σε κάθε *loop* να αρχικοποιούμε τα $dist(v, c)$ και parents.

Η παραλλαγή αυτή του αλγορίθμου έχει κόστος $\mathcal{O}(Cnm)$.

Άσκηση 2γ

- Έτσι, προκειμένου να βρούμε έναν αρνητικό κύκλο κόστους το πολύ C, μπορούμε να τρέξουμε την παραλλαγή του αλγορίθμου, με την ίδια λογική του ερωτήματος β.
- Προσέξτε τώρα, ότι μπορούμε να επεκτείνουμε την λογική του ερωτήματος β και μόλις βρούμε αρνητική απόσταση για τον αρχικό κόμβο, να σταματήσουμε και το εξωτερικό *loop* (του κόστους), να το αποθηκεύσουμε ως προσωρινό ελάχιστο κόστος και να πάμε στον επόμενο κόμβο.(Δηλαδή, αν ξεκινήσουμε από έναν κόμβο που ανήκει σε αρνητικούς κύκλους με κόστος το πολύ C, θα βρούμε αυτόν με το ελάχιστο κόστος πάλι σε χρόνο $\mathcal{O}(Cnm)$.)

Ξέρουμε ότι το μέγιστο πιθανό κόστος ενός κύκλου είναι $C_{max} = \sum w_{i,j}$ και εκτελούμε πάλι για κάθε κόμβο, άρα:

Πολυπλοκότητα

$$\mathcal{O}(C_{max} n^2 m)$$

Πρόβλημα Απόφασης
Δίνεται ένα σύστημα S με n μεταβλητές $x_1, ..., x_n$ και m ανισότητες της μορφής $x_i - x_j \leq b_{i,j}$, κάθε μία από τις οποίες έχει κόστος $w_{i,j}$.

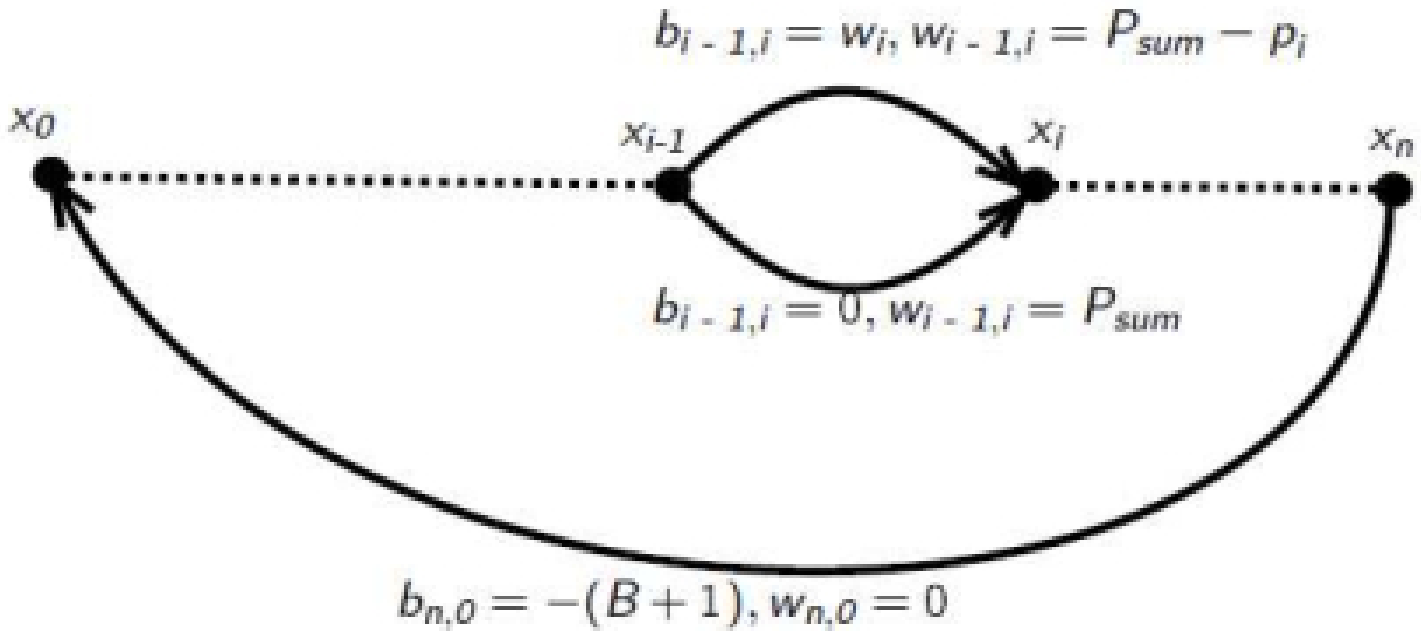
Υπάρχει μη ικανοποιήσιμο υποσύστημα S' βάρους μικρότερου ή ίσου με C ;

Στο ερώτημα 2γ, ήδη δείξαμε έναν αλγόριθμο για το πρόβλημα απόφασης. Αν δούμε την αναδρομική του σχέση, βλέπουμε ότι αυτή είναι πολύ όμοια με αυτή του Knapsack.

Αναγωγή από Knapsack:
Γενικό στιγμιότυπο από Knapsack: n στοιχεία, το i-οστό με βάρος w_i και κέρδος p_i. Υπάρχει υποσύνολό τους με συνολικό βάρος μικρότερο ή ίσο του B και συνολοκό κέρδος μεγαλύτερο ή ίσο του P ;

Θα ανάξουμε το Knapsack στο πρόβλημα εύρεσης αρνητικού κύκλου κόστους το πολύ C σε έναν γράφο της μορφής που έχουμε ορίσει για το πρόβλημα (που έχει 1-1 αντιστοιχία με το αρχικό πρόβλημα της εκφώνησης).

Έστω $P_{sum} = \sum p_i$. Για το i-οστό στοιχείο δημιουργώ μία κορυφή x_i. Συνδέω τη x_{i-1} με τη x_i με μία ακμή βάρους w_i και κόστους $c_i = P_{sum} - p_i$ και με μία ακμή βάρους 0 και κόστους P_{sum}. (Θεωρούμε και μία αρχική κορυφή x_0). Τέλος, βάζω μία ακμή από το x_n στο x_0 με βάρος $-(B+1)$ και κόστος 0. Ρωτάμε αν υπάρχει αρνητικός κύκλος κόστους το πολύ $nP_{sum} - P$.



Σχήμα: Reduction

Άσκηση 2δ

Διαισθητικά, καταλαβαίνουμε την αναγωγή ως εξής: Κάθε αρνητικός κύκλος p θα περιλαμβάνει όλους τους κόμβους (θα είναι της μορφής $(x_0, x_1, ..., x_n, x_0)$. Η επιλογή μετάβασης από τον κόμβο x_{i-1} στον x_i ισοδυναμεί με επιλογή η μη του αντικειμένου i, ανάλογα με το αν επιλέξαμε την ακμή βάρους w_i η 0 αντίστοιχα.

Έτσι, για να έχουμε αρνητικό κύκλο πρέπει το άθροισμα των βαρών των αντικειμένων που έχουμε επιλέξει να μην ξεπερνάει το αρνητικό βάρος B που έχουμε βάλει στην ακμή (x_n, x_0). Δηλαδή:

$$\sum_{(v,u)\in p} w_u \leq B$$

Τέλος, άν έχουμε βρει αρνητικό κύκλο κόστους το πολύ $P_{sum} - P$ τότε $\sum_{(v,u)\in p} c_i \leq nP_{sum} - P \Rightarrow \sum_{(v,u)\in p} P_{sum} - p_i$. Άρα:

$$\sum p_i \geq P$$

.

Άσκηση 2δ

- Έτσι βλέπουμε ότι μια λύση για το "negative cycle" πρόβλημα αντιστοιχεί σε αποδεκτή λύση για το αντίστοιχο Knapsack.
- Αντίστροφα είναι πολύ εύκολο να επιβεβαιώσουμε ότι αν το αρχικό Knapsack έχει λύση, θα έχουμε ένα ναι-στιγμιότυπο του αντίστοιχου negative cycle.
- Συνεπώς $Knapsack \leq_p negative\ cycle$.
- Έτσι, το πρόβλημα (negative cycle και συνεπώς) της εκφώνησης είναι **NP-Hard**.

Άσκηση 2δ

Προφανώς το πρόβλημα ανήκει στο NP καθώς εύκολα μπορούμε να επαληθεύσουμε αν ένα δοθέν υποσύνολο είναι μη ικανοποιήσιμο (αποτελεί αρνητικό κύκλο) και έχει βάρος μικρότερο ή ίσο του C.

Συνεπώς αφού ανήκει στο NP και είναι NP-hard, είναι **NP-complete**.

Άσκηση 3α

- Εφαρμόζουμε τον αλγόριθμο Dijkstra στο γράφημα G προκειμένου να υπολογίσουμε τις αποστάσεις $d(c_i)$ όλων των πόλεων $c_i \in C$ που διαθέτουν σταθμό φόρτισης από τον αρχικό κόμβο s.
- Δημιουργούμε ένα νέο γράφημα G' που είναι ουσιαστικά το G και στο οποίο έχουμε αντιστρέψει τη φορά όλων των ακμών. Ως αρχικό κόμβο θεωρούμε τώρα τον t.
- Εφαρμόζουμε ξανά τον Dijkstra στο G' προκειμένου να υπολογίσουμε τις αποστάσεις $d'(c_i)$ όλων των πόλεων που διαθέτουν σταθμό φόρτισης από τον αρχικό κόμβο t.
- Αναζητούμε το $min_{i \in C}(d(c_i) + d'(c_i))$
- Η πολυπλοκότητα είναι $O(mlogn)$ ή $O(m + nlogn)$.

Άσκηση 3β

- Για κάθε $c_i \in C$ υπολογίζουμε τα συντομότερα μονοπάτια από το s στο c_i και από το c_i στο t καθώς και από το c_i στο c_j για κάθε $c_i, c_j \in C$. Τέλος βρίσκουμε το συντομότερο μονοπάτι από το s στο t.
- Δημιουργούμε νέο γράφο με κόμβους όλες τις πόλεις που διαθέτουν σταθμό φόρτισης (s, t, c_i). Προσθέτω ως ακμές τα συντομότερα μονοπάτια που υπολόγισα αρκεί να μην ξεπερνούν το α. Εφαρμόζω Dijkstra στο νέο γράφο και παίρνω το συντομότερο μονοπάτι από το s στο t
- Η πολυπλοκότητα είναι $O(|C|mlogn)$ ή $O(|C|(m + nlogn))$.

Άσκηση 4α και 4β

- Για το (α): Δημιουργούμε το υπολειμματικό δίκτυο σε γραμμικό χρόνο και με ένα BFS (πάλι σε γραμμικό χρόνο) ελέγχουμε εάν υπάρχει επαυξητικό μονοπάτι. Αν δεν υπάρχει, τότε και μόνο τότε η δοθείσα ροή είναι μέγιστη.
- Για το (β): Θέλουμε να βρούμε μια εφικτή ροή στο νέο γράφημα και μετά να χρησιμοποιήσουμε τον αλγόριθμο Ford-Fulkerson ξεκινώντας από τη ροή αυτή για να φτάσουμε στη μέγιστη.
- Αν $c'_e \geq f_e$, τότε δεν χρειάζεται καμία αλλαγή. Αλλιώς θέτουμε $f'_e = c'_e$ και εκτελούμε τον ακόλουθο αλγόριθμο.
- Έστω $e = (x, y)$. Η εφικτή ροή υπολογίζεται με δύο BFS: Το πρώτο διασχίζοντας το υπογράφημα από το x στο s και το άλλο από το y στο t. Σκοπός είναι ξεκινώντας με τη μείωση κατά $f_e - c'_e$ από το x (προχωρώντας προς το s) και μετά όμοια από το y (προχωρώντας προς το t), να δημιουργηθεί μία εφικτή ροή.

Άσκηση 4β

- Κατά το BFS, έστω σε μία κορυφή v_i φτάνει νέα ροή f'_i και θέλουμε να μειώσουμε τη ροή στις προσπίπτουσες ακμές κατά k'. Μειώνεις τη ροή προς ένα γείτονα μέχρι είτε η ροή της ακμής να μηδενιστεί, είτε να έχουμε φτάσει στις k' μονάδες ροής.
- Αφού δημιουργήσουμε την εφικτή ροή f', εφαρμόζουμε τον αλγόριθμο Ford-Fulkerson. Εφόσον θα χρειαστεί προφανώς το πολύ k επαναλήψεις, η πολυπλοκότητα είναι $O(k * (V + E))$ όπου k σταθερά.

Άσκηση 4γ

- Αντιστρέφουμε όλες τις ακμές και αναζητούμε τη μέγιστη ροή. Η ιδέα είναι πως θέλουμε να δούμε πόση ροή μπορούμε να στείλουμε πίσω χωρίς να χαλάσουμε τους περιορισμούς.
- Βάζουμε νέες χωρητικότητες στο γράφημα για να λάβουμε υπόψη τους περιορισμούς: $c'_e = f_e - \ell_e$. Υπολογίζουμε τη μέγιστη ροή στο νέο γράφημα που συμβολίζουμε με h.
- Άρα η τελική ροή που είναι $g = f - h$ είναι προφανώς ελάχιστη και ισχύει επίσης
  $g_e = f_e - h_e \geq f_e - c'_e \geq f_e - f_e + \ell_e = \ell_e$ και
  $g_e = f_e - h_e \leq f_e \leq c_e$.

Άσκηση 5 3-Διαμέριση

Γενικό στιγμιότυπο του Partition: $A = w_1, ..., w_n$. Να βρεθεί διαμέριση A_1, A_2 του A ώστε $w(A_1) = w(A_2)$.
Το μετασχηματίζω σε ειδικό στιγμιότυπο του 3-Partition.
$A' = \{w_1, ..., w_n, W = \frac{w_1+...+w_n}{2}\}$.
Υποχρεωτικά, το W θα μπει σε ένα σύνολο και άρα τα άλλα δύο σύνολα είναι η διαμέριση του αρχικού προβλήματος του Partition.
Άρα NP-hard. Προφανώς μπορούμε να επαληθεύσουμε εύκολα αν μία διαμέριση είναι ισοβαρής για ένα στιγμιότυπο του 3-Partition.
Άρα ανήκει στο NP. Οπότε είναι NP-complete.

Άσκηση 5 Μακρύ Μονοπάτι

Θεώρημα:
Το LONGEST PATH είναι NP-COMPLETE

Απόδειξη:

- Πιστοποιητικό: Το μονοπάτι. Άρα LONGEST PATH $\in$ NP
- HAMILTON PATH $\leq$ LONGEST PATH

Άσκηση 5 Μακρύ Μονοπάτι-Αναγωγή

- Προσθέτουμε σε κάθε κορυφή γείτονες βαθμού 1.
- Αν έχουμε μονοπάτι Hamilton στο G, τότε θα υπάρχει μονοπάτι μήκους $|V| + 2$ στο G'
- Θέλουμε αυτό να είναι τουλάχιστον το $\frac{1}{4}$ του συνολικού αριθμού των κορυφών του G'
- Θα προσθέσουμε αρκετές καινούριες κορυφές ώστε να έχουμε $|V'| = 4|V| + 8$

Άσκηση 5 Πυκνό Γράφημα (Dense Subgraph)

Θεώρημα:
Το DENSE SUBGRAPH (DS) είναι NP-COMPLETE

Απόδειξη:

- Πιστοποιητικό: Το S. Άρα DS $\in$ NP
- DS γενίκευση του Clique ($B = \frac{k(k-1)}{2}$)

Απόδειξη

- Ανήκει στο NP: γιατί αν μας δοθεί η ανάθεση, επαληθεύουμε σε γραμμικό χρόνο ότι είναι έγκυρη.
- Είναι NP-hard: Υπάρχει πολυωνυμική αναγωγή από το 3-SAT.
- Έστω $\phi = \bigwedge_{j=1}^{m}(\ell_{j1} \vee \ell_{j2} \vee \ell_{j3})$ μία 3-CNF πρόταση για την οποία θέλουμε να ελέγξουμε αν υπάρχει ανάθεση αληθοτιμών που την ικανοποιεί (δηλαδή κάθε όρος της περιλαμβάνει τουλάχιστον ένα αληθές literal).

Άσκηση 5 Ικανοποιησιμότητα με περιορισμούς

- Κατασκευάζουμε σε πολυωνυμικό χρόνο τη λογική πρόταση $\phi' = \bigwedge_{j=1}^{m}(\ell_{j1} \vee \ell_{j2} \vee \ell_{j3} \vee z)$ όπου z καινούρια μεταβλητή που δεν εμφανίζεται στην ϕ.
- Αν υπάρχει ανάθεση που να ικανοποιεί τον περιορισμό της ϕ' τότε η ϕ είναι ικανοποιήσιμη και το αντίστροφο: Για το αντίστροφο αν η ϕ είναι ικανοποιήσιμη, έχει τουλάχιστον ένα literal αληθές σε κάθε όρο άρα η ίδια ανάθεση, αν θέσουμε επιπλέον $z = false$ ικανοποιεί τον περιορισμό μας για την ϕ'. Για το ευθύ, αν έχουμε μία ανάθεση για το ϕ' που ικανοποιεί τον περιορισμό τότε προφανώς αν $z = false$ τότε η ίδια ανάθεση ικανοποιεί και την ϕ, διαφορετικά την ικανοποιεί η αντίθετή της. (δηλ. όποια μεταβλητή έχει true κάντη false και το αντίθετο).

Απόδειξη

- Ανήκει στο NP: Ελέγχουμε τα υποσύνολα που πρέπει να ικανοποιούν το ζητούμενο σε πολυωνυμικό χρόνο.
- Είναι NP-hard: Υπάρχει πολυωνυμική αναγωγή από το Max Independent Set (MIS).
- Για κάθε κόμβο του γραφήματος G, δημιουργούμε ένα σύνολο στο S με όλες τις προσπίπτουσες ακμές του κόμβου αυτού στο γράφημα. Η σταθερά k του Set Packing είναι ίση με την αντίστοιχη του MIS. Το U είναι το σύνολο των ακμών E.

Άσκηση 5 Επιλογή Ανεξάρτητων Υποσυνόλων

- Έστω ότι έχουμε ανεξάρτητο σύνολο κορυφών με πληθάριθμο τουλάχιστον k. Κανένα ζεύγος κορυφών αυτού του συνόλου δεν θα έχει κοινή ακμή. Άρα και κανένα ζεύγος των αντίστοιχων υποσυνόλων του Set Packing δεν θα έχει κοινό στοιχείο. Δηλαδή, όλα τα υποσύνολα που θα δημιουργήσουμε θα είναι ξένα μεταξύ τους και τουλάχιστον k στο πλήθος.
- Αντίστροφα, έστω ότι έχουμε τουλάχιστον k ξένα υποσύνολα του U. Εφόσον το καθένα αντιστοιχεί σε κορυφή του αρχικού γραφήματος, θα έχουμε προφανώς στο G ίδιου πλήθους σύνολο κορυφών χωρίς κοινές ακμές, δηλαδή ένα ανεξάρτητο σύνολο με τον κατάλληλο πληθάριθμο.