Εθνικό Μετσόβιο Πολυτεχνείο
Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών
Τομέας Τεχνολογίας Πληροφορικής και Υπολογιστών
Αλγόριθμοι και Πολυπλοκότητα
Διδάσκοντες: Δημήτρης Φωτάκης, Δώρα Σούλιου, Θανάσης Λιανέας
4η Σειρά Γραπτών Ασκήσεων - Σχέδιο Λύσεων

1 Άσκηση 1

(a) Έστω ότι το συντομότερο $s-t$ μονοπάτι, p, που περνάει από κάποιον ενδιάμεσο κόμβο φόρτισης, περνάει από τον κόμβο φόρτισης $c \in C \setminus \{s, t\}$. Για να είναι βέλτιστο πρέπει το υπο-μονοπάτι του p μέχρι τον c, p_{s-c}, να είναι το συντομότερο $s-c$ μονοπάτι και το $p_{c,t}$ το ελάχιστο $c-t$ μονοπάτι.

Έτσι, αρκεί να βρούμε όλα τα ελάχιστα ζεύγη μονοπατιών $s-c$ και $c-t$, για κάθε $c \in C \setminus \{s, t\}$. Παρατηρούμε ότι αν τρέξουμε τον Dijkstra για την εύρεση του συντομότερου $s-t$ μονοπατιού, μπορούμε να κρατήσουμε τις ενδιάμεσες τιμές για τα ελάχιστ $s-c$ μονοπάτι για όλα τα c. Για να βρούμε τα ελάχιστα $c-t$ μονοπάτια, αρκεί να αντιστρέψουμε τις ακμές του γράφου και να τρέξουμε πάλι τον Dijkstra για να βρούμε το ελάχιστο $t-s$ μονοπάτι στον νέο γράφο. Κρατάμε πάλι τα ελάχιστ κόστη $t-c$ που ισοδυναμούν με το κόστος των ελάχιστων μονοπατιών $c-t$ στον αρχικό γράφο.

Συνολικά τρέχουμε τον Dijkstra δυο φορές, άρα έχουμε πολυπλοκότητα όσο και ο Dijkstra, $O(|E|+|V|\log|V|)$.

(b) Παρατηρούμε πάλι, ότι το ελάχιστο $s-t$ μονοπάτι, p, που τηρεί τον περιορισμό της μπαταρίας, αν υπάρχει, περνάει από κάποιους ενδιάμεσους κόμβους φόρτισης (το πολύ μια φορά από τον κάθε ένα, αφού έχουμε θετικά βάρη στις ακμές). Για κάθε διαδοχικούς κόμβους φόρτισης c_i, c_k από τους οποίους περνάει το p ξέρουμε, πάλι από αρχή βελτιστότητας, ότι πρέπει p_{c_i,c_k} να είναι το ελάχιστο $c_i - c_k$ μονοπάτι. Επιπλέον, αφού είναι διαδοχικοί στο p, πρέπει το κόστος του p_{c_i,c_k} να είναι μικρότερο από α.

Έτσι, για να βρούμε την λύση, μπορούμε να κάνουμε το εξής: Τρέχουμε τον Dijkstra μια φορά για κάθε $c_i \in C \setminus \{t\}$ με αφετηρία τον c_i και κρατάμε όλα τα κόστη των ελάχιστων $c_i - c_k$ μονοπατιών, για κάθε $c_k \in C \setminus \{s\}$. Έπειτα κατασκευάζουμε το γράφημα G' με σύνολο κορυφών το C και συνδέουμε τις c_i και c_k με μια ακμή βάρους ίση με το ελάχιστο $c_i - c_k$ μονοπάτι αν και μόνο αν έχει κόστος μικρότερο του α. Η λύση (αν υπάρχει) είναι το ελάχιστο $s-t$ μονοπάτι στον G' (αν υπάρχει) που την βρίσκουμε με άλλη μια εκτέλεση του αλγόριθμου Dijkstra. Έτσι, έχουμε πολυπλοκότητα $O(|C|(|E|+|V|\log|V|))$.

2 Άσκηση 2

(a) Η ιδέα είναι να αναπαραστήσουμε το $m \times n$ grid ως ένα κατευθυνόμενο γράφημα ως εξής: Για απλότητα, ας τοποθετήσουμε περιμετρικά του πλέγματος, μπλοκαρισμένους γείτονες. Για κάθε κορυφή (a, b) του πλέγματος που είναι ελεύθερη, τοποθετούμε 4 κορυφές $z_{a,b,down}, z_{a,b,up}, z_{a,b,right}, z_{a,b,left}$. Για κάθε ακμή $u \to v$ μεταξύ κορυφών του γραφήματος, θα υπάρχει μία συνάρτηση βάρους, $w : E \to \{1, 4mn+1\}$ (η οποία θα βάζει βάρος 1 αν η ακμή είναι ευθεία (straight) και $4mn+1$ αν είναι δεξιά στροφή (right)).

Θα δούμε πως τοποθετούμε ακμές και βάρη στο γράφημα. Έστω ότι το αμάξι βρίσκεται στο σημείο (a, b) και, δίχως βλάβη, ας υποθέσουμε ότι και οι 4 πιθανές επιλογές του πλέγματος είναι ελεύθερες. Κάθε μία από τις 4 κορυφές του σημείου (a, b) θα έχει in-degree και out-degree ίσους με 2 (για την ακρίβεια,θα είναι το πολύ 2 διότι ενδέχεται να έχουμε μπλοκαρισμένους γείτονες). Ασ δούμε τους έξω-βαθμούς. Οι έσω προκύπτουν επαναλαμβάνοντας τα παρακάτω βήματα.

- **Προσομοίωση κίνησης προς τα πάνω:** Για να αναπαραστήσουμε την κίνηση $(a, b) \to (a, b+1)$, τοποθετούμε ακμή $z_{a,b,down} \to z_{a,b+1,down}$ με βάρος ίσο με 1 (ευθεία). Επίσης, $z_{a,b,right} \to z_{a,b+1,down}$ με βάρος ίσο με $4mn + 1$ (δεξιά).
- **Προσομοίωση κίνησης προς τα κάτω:** Για να αναπαραστήσουμε την κίνηση $(a, b) \to (a, b - 1)$, τοποθετούμε ακμή $z_{a,b,up} \to z_{a,b-1,up}$ με βάρος ίσο με 1 (ευθεία). Επίσης, $z_{a,b,left} \to z_{a,b-1,up}$ με βάρος ίσο με $4mn + 1$ (δεξιά).
- **Προσομοίωση κίνησης προς τα δεξιά:** Για να αναπαραστήσουμε την κίνηση $(a, b) \to (a + 1, b)$, τοποθετούμε ακμή $z_{a,b,left} \to z_{a+1,b,left}$ με βάρος ίσο με 1 (ευθεία). Επίσης, $z_{a,b,down} \to z_{a+1,b,left}$ με βάρος ίσο με $4mn + 1$ (δεξιά).
- **Προσομοίωση κίνησης προς τα αριστερά:** Για να αναπαραστήσουμε την κίνηση $(a, b) \to (a-1, b)$, τοποθετούμε ακμή $z_{a,b,right} \to z_{a-1,b,right}$ με βάρος ίσο με 1 (ευθεία). Επίσης, $z_{a,b,up} \to z_{a-1,b,right}$ με βάρος ίσο με $4mn + 1$ (δεξιά).

Με τα βάρη που επιλέγουμε, τρέχοντας τον αλγόριθμο Dijkstra (έχουμε θετικά βάρη), πάντα οι ευθείες θα προτιμούνται από τις στροφές. Ο χρόνος εκτέλεσης είναι $O(mn \log(mn))$.

(b) Η ιδέα του ερωτήματος είναι να θεωρήσουμε δύο γραφήματα G_1, G_2 αντίγραφα του (α) και αν είναι $u_1 \in V(G_1) \to v_2 \in V(G_2)$ αν το αμάξι γίνεται να πάει από την u στην v του G (στο (α)) με δύο διαδοχικές δεξιές στροφές και κατάλληλα επιλεγμένο βάρος. Τρέχουμε ξανά τον ίδιο αλγόριθμο με πηγή $s_1 \in V(G_1)$ και στόχο $t_2 \in V(G_2)$. Συνολικά έχουμε $8mn$ κόμβους. Οι συνολικές ακμές είναι $2|E(G_1))| + O(8mn)$ και άρα ο χρόνος εκτέλεσης είναι $O(mn \log(mn))$.

3 Άσκηση 3

(i) Η υπόθεση του ερωτήματος είναι ότι δεν υπάρχουν κύκλοι στο γράφημα που να αυξάνουν την δύναμη του παίκτη. Παρατηρείστε ότι αν τέτοιοι κύκλοι υπήρχαν, ήταν προσβάσιμοι με αρχικό βάρος $r > 0$ και η κορυφή t ήταν προσβάσιμη από κάποια κορυφή του κύκλου, τότε θα υπήρχε ασφαλής διαδρομή $s - t$. Αρχικοποιούμε ένα reachability πίνακα d με n θέσεις με $d[v] = +\infty, \forall\, v \in V$. Επαναλαμβάνουμε την παρακάτω χαλάρωση στις (κατευθυνόμενες) ακμές του γραφήματος: Για κάθε $u \to v$, ελέγχουμε αν $d[v] > d[u] + p(v) > 0$ (οι δύο συνθήκες περιγράφουν το χειρότερο σενάριο να φτάσει ο παίκτης ζωντανός στο δωμάτιο v, αν αυτό γίνεται). Αν ναι, τότε μπορούμε να μπούμε στο δωμάτιο v από το u και η εναπομείνουσα ενέργεια να είναι $d[u] + p(v)$ (πρέπει πάντα να είναι θετική), δηλαδή ανανεώνουμε το $d[v] \leftarrow d[u] + p(v)$. Μετά το τέλος της επαναληπτικής διαδικασίας (δηλαδή $O(|V||E|)$ βήματα), ελέγχουμε αν $d[t] \in (0, \infty)$.

(ii) Σε αυτή την περίπτωση, αρκεί να ελέγξουμε αν κάποια κορυφή τέτοιων κύκλων είναι προσβάσιμη και αν η κορυφή t ήταν προσβάσιμη από κάποια κορυφή αυτού του κύκλου. Αν ναι, τότε η διαδρομή είναι ασφαλής. Μετά την ολοκλήρωση της διαδικασίας του (i), ελέγχουμε αν $d[t] \in (0, \infty)$. Αν όχι, χαλαρώνουμε τις ακμές άλλη μία φορά και αν κάποια τιμή αλλάξει, τότε υπάρχει θετικός κύκλός, προσβάσιμος από την πηγή s με αρχική ζωή r. Έτσι, μπορούμε να εντοπίσουμε τους κύκλους και να ελέγξουμε αν μπορούμε να φτάσουμε στην t (μη ενδιφερόμενοι για τα βάρη). Ο χρόνος είναι ξανά $O(|V||E|)$.

(iii) Δυαδική αναζήτηση στις τιμές και τρέχουμε τον αλγόριθμο που ελέγχει αν ο Λαβύρινθος είναι ασφαλής.

4 Άσκηση 4

Ανάγουμε το πρόβλημα σε πρόβλημα εύρεσης μέγιστης ροής. Αρχικά παρατηρούμε ότι για να κλείσει ο στρατός μια συγκεκριμένη βάση η να αιχμαλωτίσει έναν συγκεκριμένο Εξτρεμιστή, το ελάχιστο κόστος είναι να χρησιμοποιήσει τον στρατιώτη που έχει την πιο κοντική θέση (δεδομένου ότι οι στρατιώτες πάντα επιστρέφουν στις θέσεις τους μετά).

Δημιουργούμε γράφημα G με $m + k + 2$ κόμβους. Συγκεκριμένα έχουμε m κόμβους που αντιστοιχούν στους Εξτρεμιστές, k κόμβους που αντιστοιχούν στο πλήθος των βάσεων και 2 κόμβους s και t. Θέλουμε να συνδέσουμε κάθε κόμβο που αντιστοιχεί σε Εξτρεμιστή με κάθε κόμβο βάσης απο την οποία η απόστασή του είναι $\leq d$. Σε κάθε τέτοια ακμή, βάζουμε άπειρη χωριτικότητα. Επιπλέον, συνδέουμε κάθε κόμβο εξτρεμιστή με με τον κόμβο s και χωριτικότητα το ελάχιστο κόστος για τον στρατό για να εχμαλωτίσει τον αντίστοιχο Εξτρεμιστή (δηλ την απόσταση του κοντινότερου στρατιώτη σε αυτόν). Τέλος συνδέουμε κάθε κόμβο που αντιστοιχεί σε κόμβο βάσης με τον κόμβο t με χωριτικότητα το ελάχιστο κόστος για τον στρατό για να κλείσει τη βάση (δηλ την απόσταση του κοντινότερου στρατιώτη σε αυτόν).

Εύκολα παρατηρούμε ότι το $s-t$ Min Cut στον G είναι το ζητούμενο κόστος (καθώς έχουμε θέσει τα κόστη των ακμών μεταξή Εξτρεμιστών και Βάσεων να είναι άπειρα, το min cut περιέχει ακμές από το s σε κόμβους Εξτρεμιστών και από το t σε κόμβους βάσεων).

Από θεωρία, μπορούμε να βρούμε το $s - t$ Min Cut βρίσκοντας το $s - t$ Max flow με όποιον από τους γνωστούς αλγορίθμους προτιμάμε.

Το γράφημα G μπορούμε να το φτιάξουμε σε γραμμικό χρόνο, αν πρωτίστως ταξινομίσουμε τους Εξτρεμιστές, τις βάσεις και τους Στρατιώτες με βάση τη θέση τους στη γραμμή (σε $O((n+k+m)log(n+k+m)))$ όπου n το πλήθος των στρατιωτών.

5 Άσκηση 5

Θα λύσουμε και αυτό το πρόβλημα ανάγοντας το σε υποπροβλήματα εύρεσης ελάχιστης ροής. Αρχικά παρατηρούμε τα εξής: Προφανώς το ελάχιστο όριο T είναι μεγαλύτερο του 0 (εφ'όσον υπάρχουν εταιρίες με θετικό υπόλοιπο). Επιπλέον, αν $\{b_1, ..., b_k\}$ τα υπόλοιπα των εταιριών, το ελάχιστο T είναι σίγουρα μικρότερο από το μέγιστο (θετικό) υπόλοιπο, αφού τότε μπορούμε να λύσουμε όλα τα χρέη άμεσα. Δηλ $0 \leq T \leq b_{max} = \max_{i \in \{1,...,n\}}\{d_i\}$. Τέλος παρατηρούμε το εξής, αν T το ελάχιστο ικανό όριο για να ρυθμιστεί το υπόλοιπο όλων των εταιριών, τότε με κάθε όριο $y > T$ μπορούμε να ρυθμίσουμε τα υπόλοιπα ακριβώς με τον ίδιο τρόπου που το κάνουμε με όριο το T. Έτσι, αρκεί να κάνουμε δυαδική αναζήτηση στο εύρος ακέραιων τιμών από 0 εως b_{max} και να βρούμε την μικρότερη για την οποία μπορούμε να ικανοποιήσουμε τις συναλαγές.

Για να δούμε αν μπορούμε να ικανοποιήσουμε τις συναλαγές με ένα όριο y, μπορούμε να μοντελοποιήσουμε την μετακίνηση χρημάτων μεταξύ των εταιριών ως ροή σε ένα δίκτυο. Συγκεκριμένα, φτιάχνουμε ένα πλήρες γράφημα που κάθε κόμβος αντιστοιχεί σε μια από τις εταιρίες και οι ακμές έχουν χωριτικότητα y. Έπειτα, προσθέτουμε δυο ακόμα κόμβους s και t όπου ο κόμβος s συνδέεται με καθε εταιρία i για την οποία $b_i > 0$, με χωρητικότητα b_i και ο κόμβος t συνδέεται με κάθε κόμβο j για τον οποίο $b_j < 0$, με χωρητικότητα $-b_j$. Εύκολα βλέπουμε ότι αν η μέγιστη ροή $s - t$ στο γράφημα είναι μικρότερη από το άθροισμα των (θετικών) χρεών των εταιριών (δηλ $< \sum_{\{i|i\in\{1,\dots,n\}\wedge b_i>0\}} b_i$) τότε το y δεν είναι αρκετά μεγάλο για να ικανοποιηθούν οι συναλλαγές, ενώ αν είναι ίση το y αρκεί.

Έτσι, για την πολυπλοκότητα, χρειάζεται να τρέξουμε $O(\log(b_{max}))$ φορές τον αλγόριθμο μέγιστης ροής στο γράφημα που περιγράψαμε. Πχ, με αλγόριθμο push-relabel για το Max Flow, συνολικά έχουμε $O(n^3 \log(b_{max}))$

6 Άσκηση 6

(i) **Άθροισμα Υποσυνόλου κατά Προσέγγιση:** Έστω ένα instance του subset sum με θετικούς ακεραίους $w_1, \dots, w_n$ και στόχο t. Παρατηρείστε ότι είναι NP-δύσκολο να αποφανθούμε αν υπάρχει υποσύνολο που να επιτυγχάνει στόχο $t - 1$ ή t. Για κάποιο ακέραιο $x \in [1..t-1]$, κατασκευάζουμε το instance $W^\star = \{xw_1, \dots, xw_n\}$ με στόχο xt. Τότε ο black-box αλγόριθμος με είσοδο $(W^\star, xt, x)$, θα μπορούσε να αποφανθεί αν υπάρχει σύνολο $S \subseteq W^\star$ τ.ω.

$$x(t-1) \le x \sum_{i\in S} w_i \le xt$$

Το ζητούμενο έπεται.

(ii) **Τακτοποίηση Ορθογωνίων Παραλληλογράμμων:** Αναγωγή από το 2-Partition. Βασική ιδέα: Έστω τα ορθογώνια A_i με μεγέθη $\alpha_i \times \epsilon$, με $\epsilon \ll 1$ όπου μπορεί να επιλεγεί κατάλληλα. Επίσης, έστω B το μεγάλο ορθογώνιο μεγέθους $t \times 2\epsilon$. Η ιδέα είναι να επιλέξουμε $t = \frac{1}{2}\sum_i \alpha_i$ και, έτσι, αν μπορούμε να τοποθετήσουμε τα n ορθογώνια μέσα στο B τότε, λόγω του πλάτους του θα είχαμε κάνει partition το σύνολο $\{\alpha_1, \dots, \alpha_n\}$ σε δύο ισομεγέθη σύνολα. Για να λειτουργήσει αυτό, πρέπει να επιλέξουμε το ϵ κατάλληλα ώστε να μην επιτρέψουμε περιστροφές στην τοποθέτηση των A_i.

(iii) **Μέγιστη τομή με Βάρη στις Κορυφές:** Αφού κάθε weight $w_v \ge 1$, έχουμε πως, αν μπορούσαμε να βρούμε ένα partition $(S, V \setminus S)$ των κορυφών V της κλίκας με τομή τουλάχιστον βάρους , τότε

$$B \le w(S, V\setminus S) = \sum_{u\in S, v\notin S} w_u w_v = \sum_{u\in S} w_u \sum_{v\in[n]\setminus S} w_v$$

Η τελευταία ισότητα προκύπτει επειδή έχουμε κλίκα για γράφημα. Συνεχίζουμε με δύο παρατηρήσεις: Πρώτον, το παρόν πρόβλημα είναι πολυωνυμικά ισοδύναμο με το να μεγιστοποιήσουμε το βάρος της τομής (αντί να έχουμε στόχο B, δουλεύουμε στην κλάση NPO). Δεύτερον, αν $A = \sum_{i\in[n]} w_i$, το δεξί μέλος της παραπάνω ανισότητας έχει τη μορφή

$$w(S)(A - w(S)) \le A^2/4$$

Η ανισότητα αυτή προκύπτει από την μεγιστοποίηση της κοίλης συνάρτησης $f(x) = x(A - x)$, η οποία μεγιστοποιείται στην θέση $x = A/2$. Αυτό μας προτρέπει να δουλέψουμε με το πρόβλημα 2-Partition. Αν είχαμε ένα σύνολο με θετικούς ακεραίους και θέλαμε να το διαμερίσουμε σε δύο ισοβαρή υποσύνολα, τότε, από την παραπάνω ανάλυση, αρκεί να μεγιστοποιήσουμε την τομή ενός γραφήματος με κορυφές όσες και ο πληθάριθμος τους 2-Partition instance και βάρη στις κορυφές ίσα με τα στοιχεία του δοθέντος συνόλου.

(iv) **Κύκλος Hamilton κατά Προσέγγιση:** Αναγωγή από τον κλασικό κύκλο Hamilton. Έστω γράφημα $G = (V, E)$ με $V = \{v_1, \ldots, v_n\}$. Κατασκευάζουμε το $G^\star$ με $V^\star = \{\{v_1, \ldots, v_n\} \cup \{v'_1, \ldots, v'_n\}\}$ και $E^\star = E \cup \{(v_i \to v'_i)\}_{i\in[n]} \cup \{(v'_i \to v_i)\}_{i\in[n]}$. Παρατηρείστε ότι ένας κύκλος που επισκέπτεπται την v'_i, πρέπει να επισκεπτεί την κορυφή v_i δύο φορές και αυτό για κάθε $i \in [n]$.

(v) **Ικανοποιησιμότητα με Περιορισμούς:** Έστω ένα $3-$CNF instance. Κάθε όρος $r_{j1} \vee r_{j2} \vee r_{j3}$ μπορεί να επακταθεί σε ένα όρο με 4 literals, όπου το επιπλέον θα είναι η άρνηση ενός από τα υπάρχοντα. Αν μπορούσαμε να λύσουμε το δοθέν πρόβλημα, θα λύναμε και το 3SAT.

(vi) **Επιλογή Ανεξάρτητων Υποσυνόλων:** Αναγωγή από 3-dimensional matching. Μπορούμε να σκεφτόμαστε τα σύνολα S_i της εκφώνησης σαν υπερακμές ενός υπεργραφήματος με n κορυφές. Έστω ένα instance του $3DM$ με τρία σύνολα X, Y, Z και συνολικά n κορυφές και υπερακμές $\{x, y, z\} \in X \times Y \times Z$. Είναι NP-complete να βρεθεί ένα 3-dimensional matching M με $|M| \geq k$. Άρα, θα μπορούσαμε να θέσουμε ως S_i κάθε τέτοια υπερακμή και το ζητούμενο προκύπτει.