Εθνικό Μετσόβιο Πολυτεχνείο
Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών
Τομέας Τεχνολογίας Πληροφορικής και Υπολογιστών
Αλγόριθμοι και Πολυπλοκότητα
Διδάσκοντες: Δημήτρης Φωτάκης, Δώρα Σούλιου, Θανάσης Λιανέας
2η Σειρά Γραπτών Ασκήσεων - Ημ/νία Παράδοσης 3/12/2020

Άσκηση 1: Δίσκοι και Σημεία

Δίνονται n (διαφορετικά μεταξύ τους) σημεία και μία ευθεία l στο επίπεδο. Κάθε ένα από τα n σημεία βρίσκεται σε (κάθετη) απόσταση το πολύ r από την ευθεία l. Θέλουμε να καλύψουμε όλα τα n σημεία με ελάχιστο πλήθος δίσκων ακτίνας r, των οποίων τα κέντρα θα βρίσκονται πάνω στην ευθεία l. Πιο συγκεκριμένα, πρέπει να τοποθετήσουμε δίσκους ακτίνας r, κατά μήκος της ευθείας l, με το κέντρο κάθε δίσκου να ανήκει l, ώστε καθένα από τα n σημεία να καλύπτεται από τουλάχιστον έναν τέτοιο δίσκο. Αντικειμενικός μας στόχος είναι να ελαχιστοποιήσουμε το πλήθος των δίσκων που θα χρησιμοποιήσουμε. Να διατυπώσετε έναν αποδοτικό αλγόριθμο για αυτό το πρόβλημα. Να αιτιολογήσετε αναλυτικά την ορθότητα και την υπολογιστική πολυπλοκότητα του αλγορίθμου σας.

Άσκηση 2: Μεταφορά Δεμάτων

Δουλεύετε στο ταχυδρομείο και χρειάζεται να μεριμνήσετε για τη μεταφορά n ευαίσθητων πακέτων. Ο διαθέσιμος χώρος είναι περιορισμένος (βλ. black Friday, cyber Monday, και άλλα εορταστικά). Οπότε τα πακέτα πρέπει να στοιβαχθούν, το ένα πάνω στο άλλο, σε ένα μεταφορικό όχημα. Κάθε πακέτο i έχει βάρος $w_i > 0$ και αντοχή $d_i > 0$. Μια στοίβαξη είναι *ασφαλής*, αν κάθε πακέτο μπορεί να αντέξει το συνολικό βάρος των πακέτων που βρίσκονται από πάνω του (διαφορετικά, ένα πακέτο μπορεί να καταστραφεί κατά τη μεταφορά). Για παράδειγμα, η στοίβαξη $(1, \ldots, n)$ είναι ασφαλής, αν για κάθε πακέτο i, $1 \leq i \leq n-1$, ισχύει ότι $d_i \geq \sum_{j=i+1}^{n} w_j$.

(α) Ο χρόνος πιέζει! Θέλετε (εύκολα και γρήγορα) να διερευνήσετε αν υπάρχει ασφαλής στοίβαξη για τα πακέτα $(w_1, d_1), \ldots, (w_n, d_n)$. Αν υπάρχει, θα την υιοθετήσετε, διαφορετικά πρέπει να απευθυνθείτε άμεσα στον προϊστάμενό σας. Να εξετάσετε αν κάποιο από τα παρακάτω κριτήρια εγγυάται τον υπολογισμό μιας ασφαλούς στοίβαξης, εφόσον αυτή υπάρχει. Για κάθε κριτήριο, πρέπει είτε να αποδείξετε ότι οδηγεί πάντα σε ασφαλή στοίβαξη, εφοσόν αυτή υπάρχει, είτε να βρείτε ένα αντιπαράδειγμα. Για διευκόλυνση, μπορείτε να "σπάτε" τις ισοπαλίες (ως προς το κριτήριο που εφαρμόζετε) με όποιον τρόπο επιθυμείτε.

1. *Στοίβαξη με βάση το βάρος*: στοιβάζουμε τα αντικείμενα σε φθίνουσα σειρά βάρους, δηλ. $w_1 \geq w_2 \geq \cdots \geq w_n$ (το βαρύτερο στη βάση της στοίβας).
2. *Στοίβαξη με βάση την αντοχή*: στοιβάζουμε τα αντικείμενα σε φθίνουσα σειρά αντοχής, δηλ. $d_1 \geq d_2 \geq \cdots \geq d_n$ (το πιο ανθεκτικό στη βάση της στοίβας).
3. *Στοίβαξη με βάση το άθροισμα βάρους και αντοχής*: στοιβάζουμε τα αντικείμενα σε φθίνουσα σειρά βάρους και αντοχής, δηλ. $w_1 + d_1 \geq w_2 + d_2 \geq \cdots \geq w_n + d_n$.

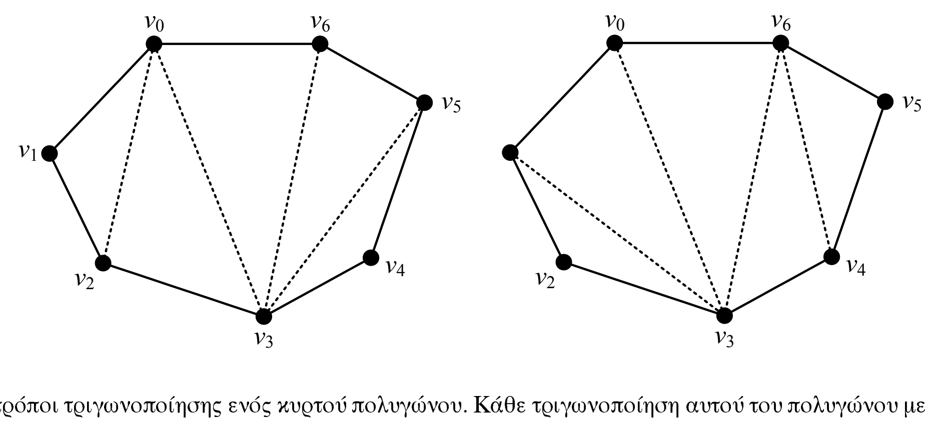
(β) Σε σας συμβαίνουν όλα! Δυστυχώς διαπιστώνετε ότι δεν υπάρχει ασφαλής στοίβαξη για όλα τα n πακέτα. Ο προϊστάμενός σας ζητάει να λάβετε υπόψη τα μεταφορικά που εισπράττει το ταχυδρομείο από κάθε πακέτο, και να υπολογίσετε ένα υποσύνολο πακέτων που μπορεί να στοιβαχθεί ασφαλώς και μεγιστοποιεί τα συνολικά έσοδα. Οπότε κάθε πακέτο i έχει βάρος $w_i > 0$, αντοχή $d_i > 0$ και δίνει κέρδος $p_i > 0$, αν επιλεγεί για μεταφορά. Να διατυπώσετε έναν αποδοτικό αλγόριθμο που υπολογίζει ένα σύνολο πακέτων που μπορούν να στοιβαχθούν ασφαλώς και μεγιστοποιούν το συνολικό κέρδος. Να αιτιολογήσετε την ορθότητα και την υπολογιστική πολυπλοκότητα του αλγορίθμου σας.

Άσκηση 3: Τριγωνοποίηση Πολυγώνου

Ως *πολύγωνο* ορίζουμε μία τμηματικά γραμμική κλειστή καμπύλη στο επίπεδο, δηλαδή μια καμπύλη που κλείνει στον εαυτό της και αποτελείται από μία ακολουθία ευθύγραμμων τμημάτων που ονομάζονται *πλευρές*. Τα σημεία τομής δύο διαδοχικών πλευρών ονομάζονται *κορυφές* του πολυγώνου. Αν το πολύγωνο είναι *απλό*, όπως υποθέτουμε, δεν τέμνει τον εαυτό του. Το σύνολο των σημείων που βρίσκονται μέσα στο απλό πολύγωνο, αποτελούν το *εσωτερικό* του πολυγώνου, το σύνολο των σημείων του πολυγώνου αποτελούν τα *όρια* του πολυγώνου, ενώ το σύνολο των σημείων γύρω από το πολύγωνο αποτελούν το *εξωτερικό* του πολυγώνου. Ένα απλό πολύγωνο είναι *κυρτό (convex)*, αν για κάθε ζευγάρι σημείων a και b που βρίσκονται στα όρια ή στο εσωτερικό του πολυγώνου, όλα τα σημεία του ευθύγραμμου τμήματος $\overline{ab}$ βρίσκονται στα όρια ή το εσωτερικό του πολυγώνου.

Μπορούμε να αναπαραστήσουμε ένα κυρτό πολύγωνο, δίνοντας τις κορυφές του με φορά αντίστροφη από αυτή των δεικτών του ρολογιού. Έτσι η ακολουθία n κορυφών $P = (u_0, u_1, \ldots, u_{n-1})$ ορίζει ένα κυρτό πολύγωνο P με n πλευρές $\overline{u_0u_1}, \overline{u_1u_2}, \ldots, \overline{u_{n-1}u_0}$.

Για κάθε ζευγάρι μη διαδοχικών κορυφών u_i και u_j, το ευθύγραμμο τμήμα $\overline{u_iu_j}$ αποτελεί μία *χορδή* του πολυγώνου. Η χορδή $\overline{u_iu_j}$ διχοτομεί το πολύγωνο σε δύο άλλα, το $P_1 = (u_i, u_{i+1}, \ldots, u_j)$ και το $P_2 = (u_j, u_{j+1}, \ldots, u_i)$. Η *τριγωνοποίηση* ενός πολυγώνου είναι ένα σύνολο χορδών T του πολυγώνου, οι οποίες χωρίζουν το πολύγωνο σε ξένα μεταξύ τους τρίγωνα (πολύγωνα τριών πλευρών). Στο Σχήμα 1 φαίνονται δύο διαφορετικοί τρόποι τριγωνοποίησης ενός κυρτού πολυγώνου. Στην τριγωνοποίηση, οι χορδές δεν τέμνονται (μπορεί όμως να εφάπτονται τα άκρα τους) και το σύνολο των χορδών T είναι μέγιστο: κάθε χορδή που δεν ανήκει στο T τέμνεται με κάποια χορδή στο T. Οι πλευρές των τριγώνων που προκύπτουν από την τριγωνοποίηση είναι είτε χορδές είτε πλευρές του πολυγώνου. Κάθε τριγωνοποίηση ενός κυρτού πολυγώνου n-πλευρών έχει $n-3$ χορδές και χωρίζει το τρίγωνο σε $n-2$ τρίγωνα. Συμβολίζουμε με $\Delta(u_iu_ju_k)$ το συνολικό μήκος των πλευρών ενός τριγώνου με κορυφές u_i, u_j και u_k που προκύπτει κατά την τριγωνοποίηση. Έχουμε ότι $\Delta(u_iu_ju_k) = \|u_iu_j\| + \|u_ju_k\| + \|u_ku_i\|$, όπου $\|xy\|$ είναι η απόσταση (Ευκλείδεια ή άλλη) των κορυφών x και y του πολυγώνου.



Σχήμα 1. Δύο τρόποι τριγωνοποίησης ενός κυρτού πολυγώνου. Κάθε τριγωνοποίηση αυτού του πολυγώνου με $n = 7$ κορυφές έχει $7 - 3 = 4$ χορδές και χωρίζει το πολύγωνο σε $7 - 2 = 5$ τρίγωνα

Δεδομένου ενός κυρτού πολυγώνου $P = (u_0, u_1, \ldots, u_{n-1})$ με n κορυφές, θέλουμε να υπολογίσουμε μια τριγωνοποίηση του P με ελάχιστο συνολικό μήκος πλευρών για τα τρίγωνα που προκύπτουν από την τριγωνοποίηση. Να διατυπώσετε έναν αποδοτικό αλγόριθμο για αυτό το πρόβλημα. Να αιτιολογήσετε αναλυτικά την ορθότητα και την υπολογιστική πολυπλοκότητα του αλγορίθμου σας.

Άσκηση 4: Τοποθέτηση Στεγάστρων (και Κυρτό Κάλυμμα)

Η διοίκηση του Πανεπιστημίου αποφάσισε να τοποθετήσει στέγαστρα στον Πεζόδρομο μπροστά στη Σχολή, για να μην μας καίει ο ήλιος το καλοκαίρι και να μην βρεχόμαστε το χειμώνα. Ο Πεζόδρομος εκτείνεται σε μία ευθεία από το σημείο $x_0 = 0$ έως το σημείο $x_f = L > 0$. Θέλουμε να καλύψουμε με στέγαστρα n συγκεκριμένα σημεία του Πεζόδρομου με συντεταγμένες $0 < x_1 < x_2 < \cdots < x_n < L$. Αυτό μπορεί να επιτευχθεί με διάφορους τρόπους, όπως π.χ., με ένα μεγάλο στέγαστρο που εκτείνεται από το σημείο x_0 έως το σημείο x_f ή με n "σημειακά" στέγαστρα που καλύπτουν μόνο τα σημεία x_i. Το κόστος για να κατασκευαστεί και να τοποθετηθεί ένα νέο στέγαστρο που εκτείνεται από κάποιο σημείο a έως κάποιο σημείο b δίνεται από την σχέση $(a-b)^2 + C$, όπου C κάποια θετική σταθερά. Σημειώνεται ότι μπορεί ένα στέγαστρο να καλύπτει μόνο ένα σημείο, δηλαδή να έχουμε $a = b$, οπότε το κόστος κατασκευής και εγκατάστασής του είναι C.

(α) Να διατυπώσετε αποδοτικό αλγόριθμο που με είσοδο τα σημεία $0 = x_0 < x_1 < x_2 \cdots < x_n < x_f = L$ και τη σταθερά C, υπολογίζει το ελάχιστο κόστος για την κάλυψη των σημείων $x_1, \ldots, x_n$ με στέγαστρα.

(β - **bonus**) Θεωρούμε ευθείες $y = ax + b$ στο $\mathbb{Q}^2$, οι οποίες περιγράφονται από το ζεύγος $(a, b) \in \mathbb{Q}^2$. Να διατυπώσετε αλγόριθμο γραμμικού χρόνου $\Theta(n + k)$, που με είσοδο ένα σύνολο n ευθειών $(a_j, b_j) \in \mathbb{Q}^2$, $j = 1, \cdots, n$, για τις οποίες ισχύει ότι $0 \geq a_1 \geq a_2 \cdots \geq a_n$, και ένα σύνολο k σημείων $x_i \in Q$, με $x_1 \leq x_2 \leq \cdots \leq x_k$, υπολογίζει τις ευθείες $(a_{j(i)}, b_{j(i)})$ που επιτυγχάνουν ελάχιστη τιμή σε καθένα από τα σημεία x_i. Πρέπει, δηλαδή, ο αλγόριθμός σας να επιστρέφει k δείκτες $j(i) = \arg\min_{j=1,\ldots,n}\{a_j x_i + b_j\}$, $i = 1, \ldots, k$. Στη συνέχεια να εξηγήσετε πως μπορούμε να χρησιμοποιήσουμε αυτόν τον αλγόριθμο για να βελτιώσουμε την υπολογιστική πολυπλοκότητα του αλγόριθμου στο (α).

Άσκηση 5: Καλύπτοντας ένα Δέντρο

Έστω δυαδικό δέντρο $T = (V, E)$ με $|V| = n$ κορυφές και ρίζα r. Για δεδομένο ακέραιο k, $1 \leq k \leq n$, θέλουμε να επιλέξουμε ένα υποσύνολο $K \subseteq V$ των κορυφών του δέντρου, με πλήθος στοιχείων $|K| = k$, που θα ελαχιστοποιεί τη μέγιστη απόσταση των κορυφών του δέντρου από τον κοντινότερο πρόγονό τους στο K. Συγκεκριμένα, θέλουμε να ελαχιστοποιήσουμε το κόστος $\mathrm{cost}(K) = \max_{v \in V}\{\mathrm{cost}(v, K)\}$, όπου $\mathrm{cost}(v, K)$ είναι η απόσταση (στο δέντρο T με ρίζα r) της κορυφής v από τον κοντινότερο πρόγονό της στο K:

$$\mathrm{cost}(v, K) = \begin{cases} 0 & \text{αν } v \in K \\ \infty & \text{αν } v = r \text{ και } r \notin K \\ \mathrm{cost}(\mathrm{par}(v), K) + 1 & \text{διαφορετικά} \end{cases}$$

Στο ορισμό του $\mathrm{cost}(v, K)$, συμβολίζουμε με $\mathrm{par}(v)$ τον "πατέρα" (άμεσο πρόγονο) της κορυφής v στο T. Παρατηρείστε ότι ο παραπάνω ορισμός απαιτεί η ρίζα r να ανήκει στο K.

(α) Να διατυπώσετε έναν αλγόριθμο δυναμικού προγραμματισμού με χρόνο εκτέλεσης $O(n^2k^2)$ που με είσοδο το δέντρο $T(V, E)$, τη ρίζα r του T, και έναν θετικό ακέραιο $k \leq n$, υπολογίζει το υποσύνολο κορυφών $K \subseteq V$, $|K| = k$, που ελαχιστοποιεί το κόστος $\mathrm{cost}(K)$. Να αιτιολογήσετε αναλυτικά την ορθότητα και την υπολογιστική του πολυπλοκότητα του αλγορίθμου σας.

(β) Να διατυπώσετε αποδοτικό αλγόριθμο που με είσοδο το δέντρο $T(V, E)$, τη ρίζα r του T και έναν θετικό ακέραιο $z \leq n$, υπολογίζει το ελάχιστο μέγεθος συνόλου $K \subseteq V$ για το οποίο $\mathrm{cost}(K) \leq z$. Να αιτιολογήσετε αναλυτικά την ορθότητα και την υπολογιστική του πολυπλοκότητα του αλγορίθμου σας. Στη συνέχεια, να εξηγήσετε πως μπορούμε να χρησιμοποιήσουμε αυτόν τον αλγόριθμο για να λύσουμε το πρόβλημα του (α) σε χρόνο σημαντικά μικρότερο του $\Theta(n^2k^2)$.