Εθνικό Μετσόβιο Πολυτεχνείο
Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών
Τομέας Τεχνολογίας Πληροφορικής και Υπολογιστών
Αλγόριθμοι και Πολυπλοκότητα
Διδάσκοντες: Δημήτρης Φωτάκης, Δώρα Σούλιου
2η Σειρά Γραπτών Ασκήσεων - Ημ/νία Παράδοσης 9/12/2019

Άσκηση 1: 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 $(1, \dots, n)$ 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 $(w_1, d_1), \dots, (w_n, d_n)$. 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 2: Αναμνηστικά

Τελικά η εργασία στο ταχυδρομείο ήταν πολύ πιεστική και όχι τόσο ενδιαφέρουσα. Μετά το παραπάνω περιστατικό, αποφασίσατε να παραιτηθείτε και να κάνετε το Γύρο του Κόσμου! Έχετε σχεδιάσει τη διαδρομή, πρόκειται να επισκεφθείτε n χώρες συνολικά, σε όλα τα μήκη και τα πλάτη της Γης. Θέλετε να φέρετε πίσω ένα αναμνηστικό από κάθε χώρα που θα επισκεφθείτε. Ως υπόδειγμα οργάνωσης, έχετε αποφασίσει ότι θα διαθέσετε C ευρώ για τα αναμνηστικά και έχετε ήδη καταγράψει κάθε αναμνηστικό που θα σας ικανοποιούσε από κάθε χώρα. Για κάθε χώρα i, έχετε ξεχωρίσει k_i αναμνηστικά. Το αναμνηστικό j από τη χώρα i έχει συναισθηματική αξία p_{ij} για σας και κοστίζει c_{ij} ευρώ. Έχετε βέβαια φροντίσει ο προϋπολογισμός σας C να επαρκεί για να αγοράσετε το φθηνότερο αναμνηστικό από κάθε χώρα, αλλά ελπίζετε ότι θα καταφέρετε να πετύχετε κάτι πολύ καλύτερο. Θέλετε λοιπόν να βρείτε έναν αλγόριθμο που υπολογίζει μια συλλογή αναμνηστικών, ένα από κάθε χώρα που θα επισκεφθείτε, η οποία θα μεγιστοποιεί τη συνολική συναισθηματική αξία και θα έχει συνολικό κόστος που δεν ξεπερνά το C. Να διατυπώσετε έναν αποδοτικό αλγόριθμο για αυτό το πρόβλημα, και να αιτιολογήσετε αναλυτικά την ορθότητα και την υπολογιστική πολυπλοκότητα του αλγορίθμου σας.

Άσκηση 3: Σοκολατάκια

Αφού τελειώσατε με τα σχέδια για το Γύρο του Κόσμου, είναι ώρα να πάρετε το γλυκάκι σας και να ξεκουραστείτε. Είστε πολύ κουρασμένος(η), πραγματικά δεν βλέπετε την ώρα να πάτε για ύπνο! Αλλά δεν μπορείτε να κοιμηθείτε, χωρίς προηγουμένως να καταναλώσετε τουλάχιστον Q σοκολατάκια (για όνειρα γλυκά), και μάλιστα σε όσο το δυνατόν μικρότερο χρονικό διάστημα. Ευτυχώς, τα σοκολατάκια δε λείπουν από το σπίτι σας, είναι άλλωστε γνωστή η αδυναμία σας σε αυτά. Έχετε n κουτιά τοποθετημένα στη σειρά και αριθμημένα από 1 μέχρι και n, από τα αριστερά προς τα δεξιά. Για να μετακινηθείτε από το ένα κουτί i στο επόμενο κουτί $i+1$ ή στο προηγούμενο κουτί $i-1$, χρειάζεστε 1 λεπτό (λόγω και της κούρασης). Τα κουτιά περιέχουν $K \leq n$ διαφορετικούς τύπους από σοκολατάκια συνολικά, και κάθε κουτί i περιέχει q_i σοκολατάκια ενός μόνο τύπου t_i. Αν ανοίξετε το κουτί i, ξέρετε καλά ότι θα καταναλώσετε, σε πρακτικά μηδενικό χρόνο, όλα τα σοκολατάκια που περιέχει, πριν συνεχίσετε στο επόμενο κουτί. Και βέβαια θα ήταν μεγάλη απογοήτευση, αν έπειτα από ένα κουτί i, ανοίγατε ένα κουτί j με σοκολατάκια του ίδιου τύπου $t_j = t_i$ με το κουτί i ή με τα ίδια ή λιγότερα σοκολατάκια $q_j \leq q_i$ από το κουτί i.

Είστε μπροστά στο κουτί p, $1 \leq p \leq n$, και θέλετε να βρείτε ποια κουτιά θα ανοίξετε απόψε, και με ποια σειρά, ώστε να καταναλώσετε τουλάχιστον Q σοκολατάκια στον μικρότερο δυνατό χρόνο, δεδομένου ότι για κάθε δύο κουτιά i και j που ανοίγετε διαδοχικά, με το i να προηγείται του j, θα πρέπει $q_i < q_j$ και $t_i \neq t_j$. Προσέξτε ότι ο χρόνος που θα χρειαστείτε καθορίζεται αποκλειστικά από τον χρόνο που χρειάζεται για να μετακινηθείτε από κάθε κουτί που ανοίγετε στο επόμενο.

Πριν πάτε για ύπνο, αποφασίζετε να λύσετε αυτό το πρόβλημα, μια για πάντα. Να διατυπώσετε έναν αποδοτικό αλγόριθμο για αυτό το πρόβλημα και να αιτιολογήσετε αναλυτικά την ορθότητα και την υπολογιστική του πολυπλοκότητα.

Άσκηση 4: Απόσταση Επεξεργασίας για Αριθμητικές Εκφράσεις

Η *απόσταση επεξεργασίας* (edit ή Levenshtein distance) $d_L(x, y)$ δυο συμβολοσειρών x και y είναι το ελάχιστο πλήθος εισαγωγών, διαγραφών και αντικαταστάσεων χαρακτήρων που χρειάζεται να γίνουν στην x ώστε αυτή να γίνει ίδια με την y. Η απόσταση επεξεργασίας $d_L(x, S)$ μια συμβολοσειράς x από μια γλώσσα S είναι η απόσταση επεξεργασίας της x από την πλησιέστερη συμβολοσειρά $y \in S$, δηλ. $d_L(x, S) = \min_{y \in S}\{d_L(x, y)\}$.

Η γλώσσα E_s των *απλών αριθμητικών εκφράσεων* ορίζεται στο αλφάβητο $\{0, 1, \ldots, 9, (,), +, -, \times, /\}$ και περιέχει κάθε συμβολοσειρά που προκύπτει από τον παρακάτω αναδρομικό ορισμό:

– Κάθε συμβολοσειρά που προκύπτει από την κανονική έκφραση $\{1|2|\cdots|9\}\{0|1|2|\cdots|9\}^*\,|\,0$ ανήκει στην E_s (δηλ. η E_s περιλαμβάνει την δεκαδική αναπαράσταση κάθε φυσικού αριθμού).
– Αν $x \in E_s$, τότε και η συμβολοσειρά $(x) \in E_s$
– Αν $x, y \in E_s$ και οι συμβολοσειρές $x+y, x-y, x\times y, x/y \in E_s$.

Να διατυπώσετε αποδοτικό αλγόριθμο που με είσοδο συμβολοσειρά $x \in \{0,1,\ldots,9,(,),+,-,\times,/\}^*$ μήκους $n \geq 1$, υπολογίζει την απόσταση επεξεργασίας $d_L(x, E_s)$ της x από την γλώσσα E_s των απλών αριθμητικών εκφράσεων. Να αιτιολογήσετε την ορθότητα και την υπολογιστική πολυπλοκότητα του αλγορίθμου σας.

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 $\Theta(n^2k^2)$.