Εθνικό Μετσόβιο Πολυτεχνείο
Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών
Τομέας Τεχνολογίας Πληροφορικής και Υπολογιστών
Αλγόριθμοι και Πολυπλοκότητα
Διδάσκοντες: Σ. Ζάχος, Δ. Φωτάκης
Συμπληρωματικές Ασκήσεις

Άσκηση 1: Φωλιασμένα Διαστήματα

Δίνονται τα μη κενά κλειστά διαστήματα φυσικών αριθμών $[s_1, t_1], [s_2, t_2], \ldots, [s_n, t_n]$. Υποθέτουμε ότι $s_1 \leq s_2 \leq \cdots \leq s_n$, ενώ κάποια από τα s_i και t_i μπορεί να είναι ίδια. Δύο διαστήματα λέγονται φωλιασμένα όταν το ένα είναι *υποσύνολο* του άλλου. Το ζητούμενο είναι να υπολογίσουμε το *πλήθος* των ζευγών φωλιασμένων διαστημάτων στο $[s_1, t_1], [s_2, t_2], \ldots, [s_n, t_n]$. Π.χ. στο $[1, 8], [2, 3], [3, 4], [5, 8], [6, 8], [7, 12]$ υπάρχουν 5 ζεύγη φωλιασμένων διαστημάτων, τα $[2, 3] \subseteq [1, 8]$, $[3, 4] \subseteq [1, 8]$, $[5, 8] \subseteq [1, 8]$, $[6, 8] \subseteq [1, 8]$, και $[6, 8] \subseteq [5, 8]$. Να διατυπώσετε ένα όσο το δυνατόν πιο αποδοτικό αλγόριθμο για αυτό το πρόβλημα. Να αιτιολογήσετε την ορθότητα και την χρονική πολυπλοκότητα του αλγορίθμου σας.

Άσκηση 2: Ενδιάμεσος Δύο Ταξινομημένων Ακολουθιών

Δίνονται δύο ταξινομημένες ακολουθίες $A = (a_1, \ldots, a_n)$ και $B = (b_1, \ldots, b_n)$, με n στοιχεία η καθεμία. Να διατυπώσετε αλγόριθμο με χρονική πολυπλοκότητα $\mathrm{O}(\log n)$ που υπολογίζει το ενδιάμεσο (median) στοιχείο της ένωσης $A \cup B$ των ακολουθιών A και B (δηλ. ζητείται το στοιχείο που θα βρίσκεται στην n-οστή θέση της ταξινομημένης ακολουθίας $A \cup B$). Για απλότητα, μπορείτε να υποθέσετε ότι το n είναι δύναμη του 2 και ότι όλα τα $2n$ στοιχεία της $A \cup B$ είναι διαφορετικά μεταξύ τους. Να αιτιολογήσετε προσεκτικά την ορθότητα και την υπολογιστική πολυπλοκότητα του αλγορίθμου.

Άσκηση 3: Το Πρόβλημα της Βιβλιοθήκης

Σε ένα φοιτητή δίνεται ένα υποχρεωτικό πρόγραμμα μελέτης για τις επόμενες T ημέρες. Με βάση το πρόγραμμα, ο φοιτητής διαβάζει ένα μόνο μάθημα κάθε ημέρα. Ο συνολικός αριθμός των μαθημάτων είναι n, $n < T$. Ο φοιτητής διαβάζει κάποια μαθήματα για περισσότερες από μία και όχι κατ' ανάγκη συνεχόμενες ημέρες (π.χ. 1η μέρα Αλγόριθμοι, 2η Γλώσσες, 3η Βάσεις, 4η Αλγόριθμοι, 5η Γλώσσες, 6η Αλγόριθμοι, 7η Αλγόριθμοι, 8η Βάσεις, 9η Βάσεις, κλπ.).

Η μελέτη κάθε μαθήματος απαιτεί το δανεισμό ενός συγκεκριμένου βιβλίου από τη βιβλιοθήκη. Ο κανονισμός δεν επιτρέπει στο φοιτητή να έχει "χρεωμένα" περισσότερα από k, $k < n$, βιβλία ταυτόχρονα. Επιπλέον, ο κανονισμός δεν επιτρέπει τον δανεισμό περισσότερων του ενός βιβλίου την ίδια ημέρα. Έτσι κάθε ημέρα που ο φοιτητής χρειάζεται ένα βιβλίο και δεν το έχει, πρέπει να πηγαίνει στη βιβλιοθήκη και να το δανείζεται (μόνο αυτό, δεν μπορεί να δανειστεί άλλα βιβλία την ίδια ημέρα). Αν μάλιστα ο φοιτητής έχει ήδη "χρεωμένα" k βιβλία, πρέπει να διαλέξει ποιο βιβλίο θα επιστρέψει. Κάθε επίσκεψη στη βιβλιοθήκη αποσπά τον φοιτητή από τη μελέτη του. Το ζητούμενο λοιπόν είναι η πολιτική επιστροφής βιβλίων που ελαχιστοποιεί τον αριθμό των επισκέψεων στη βιβλιοθήκη.

Να διατυπώσετε έναν όσο το δυνατόν πιο αποδοτικό αλγόριθμο / πολιτική επιστροφής βιβλίων που εξασφαλίζει τον ελάχιστο αριθμό επισκέψεων στη βιβλιοθήκη, και να αποδείξετε ότι ο αλγόριθμός σας πράγματι υπολογίζει τη βέλτιστη λύση. Ποιος είναι ο χρόνος εκτέλεσης του αλγορίθμου σας; Μπορείτε να συνδέσετε το πρόβλημα της βιβλιοθήκης με κάποια (σημαντική) πρακτική εφαρμογή;

Άσκηση 4: Δρομολόγηση Εργασιών

Θεωρούμε σύνολο εργασιών $A = \{1, \ldots, n\}$ που είναι υποψήφιες να εκτελεστούν σε έναν υπολογιστή. Κάθε εργασία j έχει χρόνο εκτέλεσης t_j λεπτά και προθεσμία εκτέλεσης d_j (θεωρούμε ότι οι χρόνοι εκτέλεσης και οι προθεσμίες είναι θετικοί φυσικοί αριθμοί και ότι $t_j \leq d_j$ για κάθε j). Η εκτέλεση μιας εργασίας j αποφέρει κέρδος p_j, εφόσον ολοκληρωθεί εμπρόθεσμα, και κέρδος 0, διαφορετικά.

Ο υπολογιστής μπορεί να εκτελεί μόνο μία εργασία κάθε χρονική στιγμή. Η εκτέλεση μιας εργασίας j απαιτεί την δέσμευση του υπολογιστή για χρονικό διάστημα t_j λεπτών χωρίς διακοπή. Ένα σύνολο εργασιών $F \subseteq A$ καλείται *εφικτό* αν οι εργασίες του F μπορούν να δρομολογηθούν ώστε να ολοκληρωθούν όλες εμπρόθεσμα.

(α) Να δείξετε ότι για κάθε εφικτό σύνολο εργασιών F, μια μέθοδος δρομολόγησης που εξασφαλίζει ότι όλες οι εργασίες του F ολοκληρώνονται εμπρόθεσμα είναι η δρομολόγηση τους σε αύξουσα σειρά προθεσμιών (αγνοώντας τις εργασίες εκτός F).

(b) Να διατυπώσετε έναν όσο το δυνατόν πιο αποδοτικό αλγόριθμο που υπολογίζει ένα εφικτό σύνολο εργασιών μέγιστου κέρδος. Να αιτιολογήσετε την ορθότητα και να προσδιορίσετε την χρονική πολυπλοκότητα του αλγορίθμου σας.

Άσκηση 5: Κοπή Υφασμάτων

Δίνεται ένα ορθογώνιο κομμάτι υφάσματος με διαστάσεις $X \times Y$, όπου X και Y θετικοί ακέραιοι, και μια λίστα με n προϊόντα τα οποία μπορούν να κατασκευασθούν με την χρήση του υφάσματος. Για κάθε προϊόν i γνωρίζουμε ότι χρειάζεται ένα ορθογώνιο κομμάτι υφάσματος διαστάσεων $a_i \times b_i$, και ότι η τελική τιμή πώλησης είναι c_i (τα a_i, b_i, c_i είναι θετικοί ακέραιοι, αν σας διευκολύνει, μπορείτε να υποθέσετε ότι δεν υπάρχουν προϊόντα i, j με $a_i \geq a_j$, $b_i \geq b_j$, και $c_i \leq c_j$). Κάθε προϊόν μπορεί να δημιουργηθεί σε οσαδήποτε αντίγραφα (μπορεί και κανένα). Έχουμε στη διάθεσή μας μια μηχανή που μπορεί να κόψει οποιοδήποτε ορθογώνιο κομμάτι υφάσματος σε δύο κομμάτια, είτε οριζόντια είτε κατακόρυφα (κάθε κόψιμο ακολουθεί μια ευθεία κάθετη προς τις πλευρές του υφάσματος και δεν μπορεί να διακοπεί). Να διατυπώσετε έναν όσο το δυνατόν πιο αποδοτικό αλγόριθμο που προσδιορίζει την βέλτιστη εκμετάλλευση του υφάσματος, δηλ. υπολογίζει μια στρατηγική κοπής ώστε να μεγιστοποιηθεί το συνολικό κέρδος από την πώληση των προϊόντων που προκύπτουν από τα κομμάτια του υφάσματος. Να αιτιολογήσετε την ορθότητα και να προσδιορίσετε την χρονική πολυπλοκότητα του αλγορίθμου σας.

Άσκηση 6: Αντιπροσωπεία Φορητών Υπολογιστών

Επιθυμούμε να βελτιστοποιήσουμε την λειτουργία μιας νέας αντιπροσωπείας φορητών υπολογιστών για τις επόμενες n ημέρες. Για κάθε ημέρα i, $1 \leq i \leq n$, υπάρχει (ασφαλής) πρόβλεψη d_i του αριθμού των πωλήσεων. Όλες οι πωλήσεις λαμβάνουν χώρα το μεσημέρι, και όσοι φορητοί υπολογιστές δεν πωληθούν, αποθηκεύονται. Υπάρχει δυνατότητα αποθήκευσης μέχρι S υπολογιστών, και το κόστος είναι C για κάθε υπολογιστή που αποθηκεύεται και για κάθε ημέρα αποθήκευσης. Το κόστος μεταφοράς για την προμήθεια νέων υπολογιστών είναι K ευρώ, ανεξάρτητα από το πλήθος των υπολογιστών που προμηθευόμαστε, και οι νέοι υπολογιστές φθάνουν λίγο πριν το μεσημέρι (άρα αν πωληθούν αυθημερόν, δεν χρειάζονται αποθήκευση). Αρχικά δεν υπάρχουν καθόλου υπολογιστές στην αντιπροσωπεία.

Το ζητούμενο είναι να προσδιορισθούν οι παραγγελίες (δηλ. πόσους υπολογιστές θα παραγγείλουμε και πότε) ώστε να ικανοποιηθούν οι προβλεπόμενες πωλήσεις με το ελάχιστο δυνατό συνολικό κόστος (αποθήκευσης και μεταφοράς). Να διατυπώσετε αλγόριθμο με χρόνο εκτέλεσης πολυωνυμικό στο nS για αυτό το πρόβλημα. Να αιτιολογήσετε την ορθότητα και την πολυπλοκότητα του αλγορίθμου σας.