Θεμα 1 Dijkstra

Θέμα 2 (6+(8+4) μον.). Θεωρούμε n διαστήματα $[s_1, f_1), \ldots, [s_n, f_n)$ στην ευθεία των φυσικών αριθμών (έχουμε λοιπόν ότι $s_i, f_i \in \mathbb{N}$ και $s_i < f_i$, για κάθε $1 \leq i \leq n$). Θέλουμε να επιλέξουμε κάποια από αυτά τα n διαστήματα, ώστε τα επιλεγμένα διαστήματα να μην επικαλύπτονται μεταξύ τους και να έχουν μέγιστο συνολικό μήκος. Το μήκος ενός διαστήματος $[s_i, f_i)$ είναι ίσο με $f_i - s_i$.

Παράδειγμα: Θεωρούμε $n = 5$ διαστήματα $[1,3), [2,6), [4,7), [5,8), [7,8)$. Κάποιες εφικτές λύσεις (δηλ. επιλογές διαστημάτων που δεν επικαλύπτονται μεταξύ τους) είναι οι $([1,3], [5,8))$ και $([2,6], [7,8))$, με συνολικό μήκος επιλεγμένων διαστημάτων 5, και η $([1,3), [4,7), [7,8))$, με συνολικό μήκος επιλεγμένων διαστημάτων 6. Η τελευταία συλλογή αποτελεί και την βέλτιστη λύση για το συγκεκριμένο στιγμιότυπο.

1. Να δείξετε, μέσω αντιπαραδείγματος, ότι ο άπληστος αλγόριθμος που επιλέγει, σε κάθε επανάληψη, το διαθέσιμο διάστημα $[s_i, f_i)$ με μέγιστο μήκος $f_i - s_i$ δεν οδηγεί απαραίτητα στη βέλτιστη λύση. Να κάνετε το ίδιο για τον άπληστο αλγόριθμο που επιλέγει το διαθέσιμο διάστημα $[s_i, f_i)$ με ελάχιστο χρόνο ολοκλήρωσης f_i.
2. Να διατυπώσετε αποδοτικό αλγόριθμο για το πρόβλημα της επιλογής μη επικαλυπτόμενων διαστημάτων με μέγιστο συνολικό μήκος. Να αιτιολογήσετε την ορθότητα και την υπολογιστική πολυπλοκότητα του αλγορίθμου σας.

Σημείωση: Οι 8 (από τις 12) μονάδες δίνονται για σωστό αλγόριθμο πολυωνυμικού χρόνου, οι 12 μονάδες δίνονται για σωστό αλγόριθμο με βέλτιστη υπολογιστική πολυπλοκότητα.

Θέμα 3 (7+(6+5) μον.). (α) Θεωρούμε συνεκτικό κατευθυνόμενο γράφημα $G(V, E, w)$ με n κορυφές, m ακμές, μήκος $w(e) \in \mathbb{Z}$ για κάθε ακμή $e \in E$, και αρχική κορυφή s. Για δεδομένη κορυφή $v \in V$, συμβολίζουμε με $d^{-}(v)$ το μήκος του συντομότερου $s - v$ μονοπατιού στο G που μπορεί να περιέχει **το πολύ μια ακμή αρνητικού βάρους**. Να διατυπώσετε αποδοτικό αλγόριθμο που υπολογίζει τα μήκη $d^{-}(v)$ για κάθε $v \in V$. Να αιτιολογήσετε την ορθότητα και την υπολογιστική πολυπλοκότητα του αλγορίθμου σας.

(β) Θεωρούμε συνεκτικό κατευθυνόμενο γράφημα $G(V, E, w)$ με n κορυφές, m ακμές, (μη αρνητικό) ακέραιο μήκος $w(e) \in \mathbb{N}$ για κάθε ακμή $e \in E$, και αρχική κορυφή s.

1. Έστω (σχετικά μικρός) φυσικός C τέτοιος ώστε $w(e) \leq C$ για κάθε ακμή $e \in E$. Να περιγράψετε μια υλοποίηση του αλγορίθμου Dijkstra με χρόνο εκτέλεσης $O(nC + m)$.
2. Υποθέτουμε τώρα ότι για κάποιον (σχετικά μικρό) φυσικό C, $w(e) \leq 2^{C}$ για κάθε ακμή $e \in E$. Να περιγράψετε μια υλοποίηση του αλγορίθμου Dijkstra με χρόνο εκτέλεσης $O((n + m)C)$. Να αιτιολογήσετε την ορθότητα και την υπολογιστική πολυπλοκότητα της υλοποίησης που προτείνετε.

Θέμα 4 (7+9 μον.).

1. Θεωρήστε το πρόβλημα **Subgraph Non-Isomorphism**: δίνονται γραφήματα G_1, G_2 και ζητείται να διαπιστωθεί εάν ισχύει ότι κανένα υπογράφημα του G_1 δεν είναι ισομορφικό με το G_2. Αποδείξτε ότι το **Subgraph Non-Isomorphism** είναι coNP-πλήρες (ως προς την αναγωγή πολυωνυμικού χρόνου $\leq_P$).
2. Θεωρήστε το πρόβλημα **Blue-Red Clique**: δίνεται ένα γράφημα $G(V, E)$ του οποίου κάθε κορυφή είναι χρωματισμένη με μπλε ή κόκκινο χρώμα, και ακέραιος $k > 0$, και ζητείται αν υπάρχει στο G κλίκα μεγέθους τουλάχιστον k στις κορυφές της οποίας να εμφανίζονται και τα δύο χρώματα.
   Αποδείξτε ότι το πρόβλημα **Blue-Red Clique** είναι NP-πλήρες (θεωρώντας γνωστό ότι το **Clique** είναι).

Θέμα 5 (5+5+8 μον.).

1. Θεωρούμε προβλήματα απόφασης. Ένας πιθανοτικός αλγόριθμος είναι (τύπου) *Las Vegas* αν υπολογίζει πάντα τη σωστή απάντηση, αλλά ο χρόνος εκτέλεσής του $T(n)$ είναι τυχαία μεταβλητή (π.χ., quicksort). Ένας πιθανοτικός αλγόριθμος είναι (τύπου) *Monte Carlo* αν ο χρόνος εκτέλεσής του φράσσεται ντετερμινιστικά, αλλά υπάρχει πιθανότητα λάθος απάντησης (π.χ., πιθανοτικός αλγόριθμος Min-Cut). Έστω Las Vegas αλγόριθμος A για πρόβλημα απόφασης Π με αναμενόμενο χρόνο εκτέλεσης $\mathbb{E}[T(n)] \leq f(n)$. Με βάση τον A, να διατυπώσετε αλγόριθμο Monte Carlo (για το ίδιο πρόβλημα Π) με χρόνο εκτέλεσης το πολύ $2k f(n)$ και πιθανότητα λάθους το πολύ $1/2^k$.
2. Θυμηθείτε τον 2-προσεγγιστικό αλγόριθμο για το **TSP** που είδαμε στο μάθημα, ο οποίος ξεκινάει βρίσκοντας ένα ελάχιστο συνδετικό δέντρο T και στη συνέχεια διπλασιάζει τις ακμές του T. Συμπληρώστε (συνοπτικά) την περιγραφή του αλγορίθμου και δώστε μια οικογένεια γραφημάτων για την οποία η λύση που επιστρέφει ο αλγόριθμος αυτός να είναι όσο γίνεται πιο κοντά στο διπλάσιο της βέλτιστης. Εξηγήστε την απάντησή σας.
3. Περιγράψτε τα βήματα του αλγορίθμου Knuth-Morris-Pratt στο πρόβλημα του ταιριάσματος συμβολοσειρών (string-matching). Εφαρμόστε τον αλγόριθμο στον εντοπισμό του pattern "ababaca" στο text "bacbababababacaca". Περιγράψτε μια τροποποίηση του αλγόριθμου ταιριάσματος συμβολοσειρών Knuth-Morris-Pratt στην οποία το pattern μπορεί να περιέχει οποιοδήποτε αριθμό 3 διαδοχικών ίδιων συμβόλων μπαλαντέρ =, καθένα από τα οποία ταιριάζει με τους ίδιους αυθαίρετους μεμονωμένους χαρακτήρες.
   Για παράδειγμα, το pattern ===HOC===SPOC===S:

   εμφανίζεται στο κείμενο WHBR**UUU**HOC**UUU**SPOC**UUU**SOT (μεμονωμένοι χαρακτήρες **UUU**) και
   στο κείμενο ABR**AAA**HOC**AAA**SPOC**AAA**SCADABRA (μεμονωμένοι χαρακτήρες **AAA**),
   αλλά όχι στο κείμενο FRISABAHOCAAASPOCAABSTIX.

   Υπολογίστε την πολυπλοκότητα του αλγόριθμου σας σε σχέση με m και το n, όπου m το μήκος του pattern και n το μήκος του κειμένου.