034 - Ανάλυση και Σχεδιασμός Αλγορίθμων
Προαιρετική Εργασία

Διδάσκων: Επίκ. Καθ. Παναγιώτης Πετραντωνάκης

Ομάδα 54
Ιωάννης Δημουλιός 10641
Χριστόφορος Μαρινόπουλος ΑΕΜ

Εαρινό εξάμηνο 2024

Πρόβλημα 1

Ερώτημα 1

Θέλουμε να δούμε αν είναι εφικτό ένα δρομολόγιο από την πόλη s στην πόλη t δίχως να χρησιμοποιήσουμε ακμές με e απόσταση $l_e > L$.
Επομένως, φτιάχνουμε ένα νέο γράφο $G' = (V, E')$, ο οποίος διαφέρει από τον αρχικό μόνο στις ακμές. Το νέο σύνολο ακμών E' δεν έχει τις ακμές που αναφέρονται παραπάνω, δηλαδή:

$$E' = \{e \in E \mid l_e \leq L\}$$

. Στον γράφο G' τρέχουμε τον αλγόριθμο DFS ξεκινόντας από την κορυφή s και ελέγχουμε έτσι αν υπάρχει μονοπάτι μέχρι την κορυφή t, που είναι και το ζητούμενο. Η χρονική πολυπλοκότητα του αλγορίθμου είναι $O(|V| + |E|)$, αφού προσπελαύνουμε κάθε κορυφή και κάθε ακμή του γράφου G μια φορά. ■

Ερώτημα 2

Θέλουμε να βρούμε το ελάχιστο μήκος του μακρύτερου δρόμου που πρέπει να διασχίσουμε από την πόλη s στην πόλη t. Αυτό το μήκος θα είναι και το ζητούμενο L.
Επομένως θα χρησιμοποιήσουμε τον αλγόριθμο Dijkstra ελαφρώς τροποποιημένο, ώστε το βάρος της κάθε κορυφής v να μην είναι πλέον το ελάχιστο άθροισμα των μηκών των ακμών ενός μονοπατιού από την κορυφή s έως την v, αλλά το ελάχιστο των μέγιστων μηκών των ακμών αυτών των μονοπατιών.
Για να γίνει πιο ξεκάθαρο, αν P το σύνολο των μονοπατιών από την κορυφή s μέχρι την v και, $p \in P$ ένα τέτοιο μονοπάτι, τότε ορίζουμε

$$f(p) = \max_{e \in p} l_e$$

και αν $w(\cdot)$ η συνάρτηση βάρους μια κορυφής, τότε

$$w(v) = \min_{p \in P} f(p)$$

. Ο αλγόριθμος, λοιπόν, βρίσκει και το ζητούμενο $w(t)$.
Το μόνο που χρειάζεται να αλλάξουμε στην υλοποίηση σε σύγκριση με τον κλασικό Dijkstra είναι η συνάρτηση "χαλάρωσης".
Η χρονική πολυπλοκότητα του Dijkstra, εφόσον η ουρά προτεραιότητας που απαιτείται υλοποιηθεί με δυαδικό σωρό (binary heap), είναι $O((|V| + |E|) \log |V|)$. ■

Πρόβλημα 2

Πρόβλημα 3