

Αναφορά 2ης Προγραμματιστικής Άσκησης Στο
Μάθημα
Τεχνολογίες Υλοποίησης Αλγορίθμων 2022-2023

AM = 1070263

Ον/μο = ΣΠΥΡΟ ΣΟΥΛΙ

Έτος = 6^ο

ΓΕΝΙΚΟΤΕΡΑ

Ο κώδικας παράγει έναν κατευθυνόμενο γράφο, εκτελεί μετατροπή από τη βιβλιοθήκη LEDA στη βιβλιοθήκη Boost Graph Library και εφαρμόζει τον αλγόριθμο Bellman-Ford χρησιμοποιώντας τόσο τις προσαρμοσμένες όσο και τις παρεχόμενες από τη βιβλιοθήκη συναρτήσεις. Αρχικά, χρησιμοποιώ βιβλιοθήκες όπως η boost/graph και η LEDA/graph για με γραφήματα. Ορίζω έναν κατευθυνόμενο γράφο χρησιμοποιώντας μια αναπαράσταση λίστας γειτνίασης (adjacency_list) και εισάγω ιδιότητες για τις ακμές χρησιμοποιώντας το πρότυπο ιδιοτήτων, συμπεριλαμβανομένης της ιδιότητας edge_weight_t. Επιπλέον, δηλώνω διάφορους τύπους όπως Vertex, Edge, Edgelterator και VertexIterator για το γράφημα που θα με βοηθήσουν αργότερα. Η συνάρτηση CopyLedaGraphToBoostGraph είναι υπεύθυνη για την αντιγραφή ενός κατευθυνόμενου γράφου LEDA

(`leda::graph`) σε έναν κατευθυνόμενο γράφο Boost (`DirectedGraph`), μαζί με τα σχετικά βάρη ακμών. Η συνάρτηση `BellmanFord` εφαρμόζει τον αλγόριθμο Bellman-Ford στον κατευθυνόμενο γράφο προκειμένου να προσδιορίσω τα συντομότερα μονοπάτια από μια καθορισμένη αρχική κορυφή. Χρησιμοποιώ χάρτες ιδιοτήτων για την πρόσβαση στα βάρη των ακμών και την ενημέρωση του κόστους των κόμβων. Η συνάρτηση επιστρέφει `true` αν δεν υπάρχει κύκλος αρνητικών βαρών στο γράφημα και `false` σε αντίθετη περίπτωση. Η κύρια συνάρτηση αρχικοποιεί τις μεταβλητές και λαμβάνει δεδομένα από τον χρήστη για τον τύπο του γράφου (πλέγμα ή τυχαίο) και τον αριθμό των κόμβων. Δημιουργεί είτε ένα γράφημα πλέγματος είτε ένα τυχαίο γράφημα χρησιμοποιώντας τη βιβλιοθήκη LEDA και αναθέτει τυχαία βάρη στις ακμές. Στη συνέχεια, αντιγράφει το γράφημα LEDA στο γράφημα Boost χρησιμοποιώντας τη συνάρτηση `CopyLedaGraphToBoostGraph`. Επιπλέον, παράγει μια τυχαία αρχική κορυφή και εφαρμόζει τον αλγόριθμο Bellman-Ford χρησιμοποιώντας τόσο την συνάρτηση που ορίζει ο χρήστης (`BellmanFord`) όσο και την υλοποίηση της βιβλιοθήκης Boost Graph (`bellman_ford_shortest_paths`). Μετρά και εμφανίζει τους χρόνους εκτέλεσης αυτών των αλγορίθμων.

ΕΙΔΙΚΟΤΕΡΑ

Στην περίπτωση των grid γραφημάτων. Τα γραφήματα αυτά ικανοποιούν τις συνθήκες για τις κατευθύνσεις των ακμών στα τεταρτημόρια 1, 2 και 4. Για το 3ο τεταρτημόριο, η κατεύθυνση των ακμών καθορίζεται τυχαία και αλλάζει πριν την αντιγράψει το γράφημα στην υλοποίηση της Boost. Χρησιμοποιώντας την συνάρτηση `div` της

C++, ο αριθμός Index του κόμβου στο γράφημα μετατρέπεται σε μορφή σημείου όπου ο αριθμός division αντιστοιχεί στην γραμμή του κόμβου στο γράφημα, ενώ ο αριθμός remainder αντιστοιχεί στην στήλη του κόμβου στο γράφημα. Για την αναπαράσταση του $+\infty$, έχει χρησιμοποιηθεί η τιμή INT_MAX της C++. Για την αναπαράσταση της τιμής nil, έχει χρησιμοποιηθεί η τιμή NULL της C++. Στην περίπτωση, όπου στο γράφημα υπάρχει αρνητικός κύκλος, δεν καλείται η συνάρτηση Bellman Ford της βιβλιοθήκης Boost. Τα γραφήματα δημιουργούνται με την χρήση των συναρτήσεων της LEDA και μετά την ολοκλήρωση της δημιουργίας αντιγράφονται στην υλοποίηση της Boost.

Πειραματική Αξιολόγηση

Rand Graph	Υλοποίηση Αλγορίθμου	Υλοποίηση LEDA	Υλοποίηση Boost
1000 Κόμβοι	35.35 seconds	0.0200005 seconds	Αρνητικός Κύκλος
2000 Κόμβοι	329 seconds	0.0400505 seconds	Αρνητικός Κύκλος
4000 Κόμβοι	1020 seconds	0.0700076 seconds	Αρνητικός Κύκλος

Grid Graph	Υλοποίηση Αλγορίθμου	Υλοποίηση LEDA	Υλοποίηση Boost
100	50.47 seconds	0 seconds	0.00999832 seconds
200	982.52 seconds	0.0100099 seconds	0.099756 seconds
300	4777.44 seconds	0.05 seconds	1.03 seconds

