2025

Προγραμματιστική Άσκηση 2

Γρύλλιας Γεράσιμος 1084651

Υλοποίηση Αλγορίθμου Bellman-Ford

Αρχείο bellmanFord.cpp

Update_pred(...)

Αναδρομική συνάρτηση που ενημερώνει τον πίνακα pred με βάση τις κορυφές που ανήκουν στο σύνολο R και έχουν επιτευχθεί μέσω DFS από κορυφές του συνόλου U. Πρόκειται για ένα μεταγενέστερο βήμα επεξεργασίας των αποτελεσμάτων του Bellman-Ford.

bellmanFord(...)

Κύρια συνάρτηση που υλοποιεί τον αλγόριθμο. Περιλαμβάνει:

Αρχικοποίηση αποστάσεων (dist) και προκατόχων (pred)

Επεξεργασία κορυφών μέσω ουράς (Q) σε φάσεις (μετρώντας μέχρι n φάσεις)

Έλεγχο για ενημέρωση τιμών dist και pred σε κάθε ακμή

Post-processing:

Αφαίρεση ακμών που δεν χρησιμοποιήθηκαν

Αναδρομική DFS για εύρεση του συνδεδεμένου υποσυνόλου R

Επαναφορά των ακμών στον γράφο

Ενημέρωση των pred για τις κορυφές που είναι προσβάσιμες από το U εντός του R

Αρχείο main.cpp:

Το πρόγραμμα δημιουργεί δύο τύπους γράφων:

Τυχαίος γράφος μεγέθους 1000, 4000 ή 8000 κόμβων, με τυχαία βάρη ακμών.

Πλέγματικός γράφος (Grid Graph) μεγέθους 100x100, 200x200 ή 300x300, όπου οι ακμές δημιουργούνται με συγκεκριμένη λογική χωρισμού σε τρεις περιοχές και ορισμένα βάρη ακμών είναι αρνητικά.

Αρχικά, ο γράφος κατασκευάζεται με χρήση LEDA. Έπειτα μετατρέπεται σε γράφο Boost για να εφαρμοστεί ο Bellman-Ford αλγόριθμος και στις δύο βιβλιοθήκες, καθώς και σε δική του υλοποίηση. Η μετατροπή μεταξύ LEDA και Boost γραφικών δομών είναι λειτουργική και υλοποιείται στην συνάρτηση LEDA_Graph_To_Boost(...)

Οι χρόνοι εκτέλεσης καταγράφονται και συγκρίνονται για:

Την υλοποίηση Bellman-Ford του χρήστη

Την αντίστοιχη συνάρτηση της LEDA

Την αντίστοιχη συνάρτηση της Boost Graph Library

Επίσης, γίνεται έλεγχος ύπαρξης αρνητικών κύκλων.

Συμπέρασμα

Το πρόγραμμα επιτυγχάνει την ολοκληρωμένη σύγκριση διαφορετικών υλοποιήσεων του αλγορίθμου Bellman-Ford σε μεγάλους και σύνθετους γράφους, επιβεβαιώνοντας τη λειτουργικότητα και αξιοπιστία τους, ενώ παράλληλα αναδεικνύει τη σημασία της επιλογής κατάλληλων δομών και εργαλείων ανάλογα με τις ανάγκες.

Πειραματική Αξιολόγηση

Τυχαία γραφήματα με n = 1000

n=1000	Run 1	Run 2	Run 3	Run 4	Run 5
My Func:	1.09s	0.9s	1.4s	1.29s	1.2s
LEDA:	0.95s	0.89s	1.07s	1.09s	1.05s
Neg Cycle?	Yes	Yes	Yes	Yes	Yes
Boost:	--	--	--	--	--

Μέσος Όρος My Func: 1.176 sec

Μέσος Όρος LEDA: 1.01 sec

Τυχαία γραφήματα με n = 4000

n=4000	Run 1	Run 2	Run 3	Run 4	Run 5
My Func:	61.31s	62.79s	74.82s	76.03s	77.77s
LEDA:	42.21s	42.95s	46.77s	37.57s	41.36s
Neg Cycle?	Yes	Yes	Yes	Yes	Yes
Boost:	--	--	--	--	--

Μέσος Όρος My Func: 70.54 sec

Μέσος Όρος LEDA: 42.172 sec

Τυχαία γραφήματα με n = 8000

n=8000	Run 1	Run 2	Run 3	Run 4	Run 5
My Func:	245.35s	222.03s	223.69s	332.52s	258.67s
LEDA:	187.05s	163.77s	169.12s	180.7s	177.47s
Neg Cycle?	Yes	Yes	Yes	Yes	Yes
Boost:	--	--	--	--	--

Μέσος Όρος My Func: 256.45 sec

Μέσος Όρος LEDA: 175.622 sec

Grid γραφήματα με n = 100

n=100	Run 1	Run 2	Run 3	Run 4	Run 5
My Func:	0.6s	0.78s	0.26s	0.84s	0.76s
LEDA:	1.16s	1.43s	0.5s	1.54s	1.38s
Neg Cycle?	Yes	Yes	Yes	Yes	Yes
Boost:	--	--	--	--	--

Μέσος Όρος My Func: 0.648 sec

Μέσος Όρος LEDA: 1.202 sec

Grid γραφήματα με n = 200

n=200	Run 1	Run 2	Run 3	Run 4	Run 5
My Func:	8.64s	8.26s	5.88s	8.01s	8.19s
LEDA:	15.51s	15.53s	10.62s	14.05s	14.26s
Neg Cycle?	Yes	Yes	Yes	Yes	Yes
Boost:	--	--	--	--	--

Μέσος Όρος My Func: 7.796 sec

Μέσος Όρος LEDA: 13.994 sec

Grid γραφήματα με n = 300

n=300	Run 1	Run 2	Run 3	Run 4	Run 5
My Func:	37.51s	73.4s	79.24s	94.11s	79.22s
LEDA:	31s	55.16s	64.33s	83.86s	70.24s
Neg Cycle?	Yes	Yes	Yes	Yes	Yes
Boost:	--	--	--	--	--

Μέσος Όρος My Func: 72.696 sec

Μέσος Όρος LEDA: 60.918 sec

Σχολιασμός Πειραματικών Αποτελεσμάτων

Από τα πειραματικά αποτελέσματα παρατηρούμε ότι:

Στα τυχαία γραφήματα (n = 1000, 4000, 8000), η υλοποίησή μου είναι σταθερά πιο αργή από την υλοποίηση της LEDA, με τη διαφορά να γίνεται σημαντικά μεγαλύτερη όσο αυξάνεται το μέγεθος του γράφου. Αυτό οφείλεται πιθανότατα σε βελτιστοποιήσεις που εφαρμόζει η LEDA στο χειρισμό δομών δεδομένων και στις εσωτερικές υλοποιήσεις των αλγορίθμων.

Σε όλα τα τυχαία γραφήματα ανιχνεύθηκε αρνητικός κύκλος, γεγονός που καθιστά αδύνατη τη σύγκριση με τη Boost, η οποία δεν υποστηρίζει αυτή την περίπτωση.

Στα γραφήματα τύπου πλέγματος (grid), η εικόνα είναι διαφοροποιημένη. Η υλοποίησή μου εμφανίζει καλύτερους χρόνους εκτέλεσης από τη LEDA για μικρότερες τιμές του n (n = 100, 200), ενώ για n = 300 η LEDA παρουσιάζει καλύτερη σταθερότητα και τελικά μικρότερο μέσο χρόνο εκτέλεσης.

Και στα grid γραφήματα εντοπίστηκε αρνητικός κύκλος, επομένως και πάλι η Boost δεν περιλαμβάνεται στη σύγκριση.

Συνολικά, η LEDA εμφανίζει καλύτερη κλιμάκωση και βελτιστοποιημένη απόδοση σε μεγάλα και πυκνά γραφήματα, ενώ η δική μου υλοποίηση είναι ανταγωνιστική ή καλύτερη σε μικρότερα ή αραιότερα γραφήματα.