Στοιχεία Φοιτητή

Ονοματεπώνυμο: Δημήτριος Κωστορρίζος

Αριθμός Μητρώου: 1054419

Έτος Σπουδών: E’

Θέμα Τελικής Άσκησης: Eυρετικές μέθοδοι εύρεσης συντομότερων διαδρομών ALT

Περιγραφή Υλοποίησης

Αρχικά, αποφάσισα τι είδους γραφήματα θα χρησιμοποιήσω για να αξιολογήσω την συνάρτηση ALT. Επέλεξα να χρησιμοποιήσω τυχαία γραφήματα, γραφήματα τύπου πλέγματος(grid) και πλήρη γραφήματα. Για αν δημιουτγήσω τα γραφήματα αυτά χρησιμοποίησα τις συναρτήσεις δημιουργίας γραφημάτων της LEDA, random\_graph, grid\_graph και complete\_graph.

Για τα γραφήματα σε LEDA, επέλεξα την κλάση graph() με διπλής ακρίβειας αριθμητική τιμή(double) ως κόστη ακμών.

Για τα γραφήματα σε Boost, επέλεξα την κλάση adjacency\_list() με διπλής ακρίβειας αριθμητική τιμή(double) ως κόστη ακμών. Η υλοποίηση της ιδιότητας «κόστος ακμής», έγινε χρησιμοποιώντας την εντολή: **property<edge\_weight\_t, double>** . Για την υλοποίηση των κόμβων του γραφήματος, χρησιμοποιήθηκε το vertex\_descriptor του adjacency\_list, ενώ για την υλοποίηση των ακμών του γραφήματος χρησιμοποιήθηκε το edge\_descriptor. Προκειμένου να έχω την δυνατότητα να διατρέχω την Boost υλοποίηση των γραφημάτων, χρησιμοποίησα το edge\_iterator ως iterator για να διατρέχω τις ακμές του γραφήματος, out\_edge\_iterator ως iterator για να διατρέχω τις εξερχόμενες ακμές ενός κόμβου του γραφήματος και vertex\_iterator ως iterator για να διατρέχω τους κόμβους του γραφήματος. Τέλος, χρησιμοποιήθηκε η εντολή **property\_map<DirectedGraph, edge\_weight\_t>** ώστε να ορισθεί η σχέση μεταξύ του γραφήματος και των τιμών κόστους για τις ακμές που περιέχει.

Για τα τυχαία γραφήματα, επέλεξα να μην περιέχουν παράλληλες, αντί-παράλληλες και αυτό-κυκλικές ακμές. Ως αριθμό ακμών στα τυχαία γραφήματα, όρισμα την τιμή N \* (N - 1), όπου Ν ο αριθμός των κόμβων στο γράφημα, η οποία αποτελεί τον μέγιστο αριθμό ακμών.

Για τα πλήρη γραφήματα και τα γραφήματα τύπου πλέγματος, χρησιμοποιήθηκε η προκαθορισμένο υλοποίηση της συνάρτησης της LEDA.

Για το κόστος της κάθε ακμής, ορίστηκε μία τυχαία τιμή μεταξύ του 1 και του 101.

Οι μετρήσεις για τα προαναφερθέντα γραφήματα εκτελέστηκαν για τα εξής πλήθη κόμβων: 10, 100, 1000 κόμβοι. Προσπάθησα να εκτελέσω τις δοκιμές για 10000 και 100000 κόμβους, αλλά ο Διογένης δεν επέτρεπε την δέσμευση περαιτέρω μνήμης, οπότε και δεν εκτελέστηκαν μετρήσεις για τους συγκεκριμένους αριθμούς κόμβων.

Έχοντας δημιουργήσει τα γραφήματα, χρησιμοποίησα μία συνάρτηση για να «αντιγράψω» τα γραφήματα από την υλοποίηση σε LEDA, στην υλοποίηση σε Boost. Η συγκεκριμένη συνάρτηση δημιουργεί ένα γράφημα σε Boost με ίσο αριθμό κόμβων με το αντίστοιχο γράφημα σε LEDA. Έπειτα, κάθε ακμή του γραφήματος σε LEDA, δημιουργείται στην Boost υλοποίηση του γραφήματος, με βάση τον αρχικό κόμβο, τον τελικό κόμβο και το κόστος της ακμής. Μετά την ολοκλήρωση της αντιγραφής των ακμών, ο δείκτης της Boost υλοποίησης του γραφήματος ενημερώνεται ώστε να επιστραφεί ως out παράμετρος.

Για την υλοποίηση της συνάρτησης ALT σε Boost, χρησιμοποιήθηκε ως βάση η υλοποίηση της ευρετικής αναζήτησης A\*. Αρχικά, ορίζονται τα κόστη των ακμών, καθώς και δύο χάρτες που περιέχουν ζευγάρια κόμβων-αριθμιτικών τιμών διπλής ακρίβειας(double). Ο πρώτος χάρτης θα περιέχει τα κόστη ανακάλυψης των κόμβων, ενώ ο δεύτερος θα περιέχει τα ευρετικά κόστη ανακάλυψης των κόμβων. Οι δύο αυτοί χάρτες αρχικοποιούνται στην μέγιστη αριθμητική τιμή διπλής ακρίβειας, DBL\_MAX. Για την υλοποίηση της ουράς των «ανοιχτών» κόμβων χρησιμοποιήθηκε ο τύπος **std::vector** ενώ για την υλοποίηση των χαρτών ο τύπος **std::map**.

Έπειτα, ορίζονται οι αρχικές τιμές ευρετικού κόστους και κόστους ανακάλυψης για το αρχικό κόμβο και ο κόμβος αυτός τοποθετείται στο διάνυσμα «ανοιχτών» κόμβων. Μέχρις ότου το διάνυσμα των ανοιχτών κόμβων να είναι άδειο, γεγονός που δηλώνει ότι ο τελικός κόμβος δεν μπορεί να προσεγγιστεί, εκτελέιται ο παρακάτω αλγόριθμος.

Grid Graphs

|  |  |  |
| --- | --- | --- |
| Αριθμός Κόμβων / Μέθοδος | User Defined ALT | Leda Compute Shortest Path |
| 10 | 0 seconds | - |
| 100 | 0 seconds | - |
| 1000 | 1.78 seconds | - |

Complete Graphs

|  |  |  |
| --- | --- | --- |
| Αριθμός Κόμβων / Μέθοδος | User Defined ALT | Leda Compute Shortest Path |
| 10 | 0 seconds | 0 seconds |
| 100 | 0 seconds | 0 seconds |
| 1000 | 10.34 seconds | 0.11 seconds |

Random Graphs

|  |  |  |
| --- | --- | --- |
| Αριθμός Κόμβων / Μέθοδος | User Defined ALT | Leda Compute Shortest Path |
| Κόμβοι 10, Ακμές 90 | 0 seconds | 0 seconds |
| Κόμβοι 100, Ακμές 9.900 | 0 seconds | 0 seconds |
| Κόμβοι 1000, Ακμές 909.000 | 368.91 seconds | 0.29001 seconds |