Αναφορά Εργασίας

Υλοποίηση

Η εργασία αφορούσε την υλοποίηση δυο αλγόριθμων σχετικά με την διατήρηση των bridge-connected components και biconnected components σε μη κατευθυνόμενα γραφήματα. Αρχικά παρουσιάζεται ο αλγόριθμος για bridge-connected components.

Για την υλοποίηση του ορίστηκε το αντικείμενο του κόμβου του γραφήματος Node, όπου διατηρούνταν σχετικές πληροφορίες που χαρακτήριζαν τους κόμβους. Συγκεκριμένα ένα αναγνωριστικό για την ονομασία του κόμβου, ένα πεδίο label όπου κρατούσε το όνομα του label των στρογγυλών κόμβων του γραφήματος, ένα πεδίο που δήλωνε τον τύπο του κόμβου σε κυκλικό ή τετράγωνο, ένα πεδίο Node* όπου ο κόμβος κρατούσε μια αναφορά για τον πατέρα του κόμβου και ένα πεδίο vector<Node*> όπου ο κόμβος διατηρεί την λίστα των παιδιών του κόμβου. Η ονομασία των κόμβων για να είναι μοναδική έγινε μέσω ένα static counter όπου αυξανόταν κάθε φορά που δημιουργούταν ένας νέος κόμβος και όριζε το όνομα του.

Αρχικά για την προσθήκη νέων τετράγωνων κόμβων δημιουργούμε δύο κόμβους, έναν τετράγωνο και ένα στρογγυλό. Για την διατήρηση των κόμβων χρησιμοποιούμε δομή δεδομένων τύπου vector για να καλύψουμε τις δυναμικές ανάγκες του αλγόριθμου σχετικά με την προσθήκη καινούργιων κόμβων. Για την ευκολότερη διαπίστωση αν ένας κόμβος ανήκει στο ίδιο bridge-connected

component υλοποιούμε ένα Disjoint Data Set Structure μέσω ενός unordered map και των συναρτήσεων του Union, Find και make_set για να κάνουν ένωση δύο στοιχείων, εύρεση ενός στοιχείο και προσθήκη στοιχείου στον πίνακα.

Αν δοθεί σαν όρισμα κάποιο label πέρα του μηδενός τότε δημιουργούμε έναν κυκλικό κόμβο και τον προσθέτου στο vector του γραφήματος και στο Disjoint Set. Έπειτα προχωράμε στη σύνδεση των δύο μέσω της συνάρτησης link η οποία ενώνει αυτός τους δύο κόμβους με την σχέση πατέρα παιδιού και ενημερώνει τα αντίστοιχα πεδία του καθενός. Μετά εφαρμόζουμε την συνάρτηση union για να ενώσουμε αυτά τα δύο subsets σε ένα.

Για την εύρεση του μονοπατιού μεταξύ δύο κόμβων χρησιμοποιούμε την συνάρτηση findpath. Αν οι δύο κόμβου που ψάχνουμε το μονοπάτι τους είναι ίδιοι τότε επιστρέφουμε μόνο των γονέα τους. Αλλιώς ξεκινώντας από τους γονείς των κόμβων που δόθηκαν σαν ορίσματα κρατάμε μια λίστα για τον κάθε ένα που κρατάμε τους κόμβους που βρίσκουμε ακολουθώντας τους γονιούς κάθε κόμβου μέχρι να φτάσουμε στην ρίζα.

Έπειτα για κάθε κόμβο του ενός μονοπατιού που αποθηκεύσαμε ελέγχουμε αν εμφανίζεται στο μονοπάτι του δεύτερου. Μόλις βρεθεί αυτός ο κόμβος τον κρατάμε ως τον κοντινότερο κοινό πρόγονο των δύο αυτών κόμβων. Αποθηκεύουμε σε μια ξεχωριστή λίστα το μονοπάτι μιας λίστας μέχρι το σημείο που βρέθηκε ο κοντινότερος κοινός πρόγονος και έπειτα προσθέτουμε στην ίδια λίστα την δεύτερη λίστα μέχρι να συναντήσουμε πάλι τον κοινό κόμβο και επιστρέφουμε το μονοπάτι αυτό.

Μια ακόμα συνάρτηση απαραίτητη για την διατήρηση της δομής είναι η evert η οποία κάνει τον κόμβο που παίρνει σαν όρισμα την ρίζα του δέντρο στο οποίο

βρίσκεται. Αρχικά για να προχωρήσουμε ελέγχουμε αν ο κόμβος που μας δόθηκε ήταν ήδη ρίζα του του δέντρου του ελέγχοντας αν έχει πατέρα. Η απώλεια πατέρα μας βεβαιώνει ότι είναι η ρίζα. Εδώ θα μπορούσε να χρησιμοποιηθεί και η συνάρτηση find, η οποία θα επέστρεφε την ρίζα του δέντρο που βρίσκεται ο κόμβος, συγκρίνοντας το αποτέλεσμα της με το όνομα του κόμβου. Αν είναι ίδιο τότε αυτός ο κόμβος είναι η ρίζα.

Όταν δεν είναι η ρίζα τότε διατρέχουμε όλα τα παιδιά του γονέα του στρογγυλού κόμβου (έχουμε δώσει σαν όρισμα τετράγωνο κόμβο οπότε ρίζα θα γίνει ο κυκλικός κόμβος που είναι πατέρας του) και διαγράφουμε τον στρογγυλό κόμβο από την λίστα των παιδιών του. Έπειτα χρησιμοποιούμε την συνάρτηση link για να συνδέσουμε την καινούργια ρίζα με τον πρώην πατέρα της σαν παιδί της ανανεώνοντας και το disjoint set που έχουμε ώστε όλα τα μέλη του set να δείχνουν στην ίδια ρίζα.

Η εύρεση της ταυτότητας του block όπου ανήκει ένα κόμβος γίνεται με την συνάρτηση find block. Η συνάρτηση κάνει χρήση της ήδη ορισμένη συνάρτησης find path όπου της δίνουμε σαν όρισα δύο φορές τον ίδιο κόμβο για να μας επιστρέψει τον πατέρα του κόμβου. Από αυτών τον κόμβο αποκτούμε πρόσβαση στο πεδίο label, που είναι το label που ανήκει ο κόμβος αυτός.

Μια άλλη βασική συνάρτηση είναι η condense path, η οποία ενώνει τους κόμβους ενός μονοπατιού σε ένα καινούριο κόμβο και παίρνει σαν παιδία του όλα τα παιδιά που είχαν οι κυκλικοί κόμβοι σε αυτό το μονοπάτι. Αρχικά δημιουργούμε ένα νέο κυκλικό κόμβο με το label που δώσαμε σαν όρισμα στην συνάρτηση. Έπειτα ελέγχουμε αν το μονοπάτι μεταξύ των δύο κόμβων που μας δόθηκε έχει την ρίζα ως τον πλησιέστερο κοινό κόμβο του. Αν ναι τότε και ο κόμβος που μόλις

δημιουργήσαμε θα γίνει η καινούργια η ρίζα του δέντρου. Αλλιώς ο καινούργιος κόμβος που δημιουργήσαμε θα γίνει παιδί του πατέρα του πλησιέστερου κοινού κόμβου του μονοπατιού. Στην συνέχεια προσθέτουμε τον κόμβο που δημιουργήσαμε στο disjoint set και ξεκινάμε να διατρέξουμε τους κόμβους του μονοπατιού. Για κάθε κόμβο διατρέχουμε την λίστα των παιδιών του και κάθε τετράγωνο κόμβο τον συνδέουμε με τον καινούργιο κόμβο μας.

Τέλος έχουμε την συνάρτηση με την οποία υλοποιούμε τις ακμές στην δομή δεδομένων μας. Αρχικά ελέγχουμε σε ποια από τις δύο περιπτώσεις βρισκόμαστε, αν δηλαδή οι δύο κόμβοι βρίσκονται στο ίδιο component ή όχι.

Αν βρίσκονται στο component όπου το ελέγχουμε χρησιμοποιώντας την συνάρτηση find στους δύο κόμβους. Δημιουργούμε ένα καινούργιο label και μετά εφαρμόζουμε την συνάρτηση condense path σε αυτούς τους δυο κόμβους επειδή η καινούργια ακμή δημιουργεί ένα κύκλο στο γράφημα οδηγώντας στην σύμπτυξη των κόμβων ανάμεσα στο μονοπάτι τους.

Αν όμως βρίσκονται σε δύο διαφορετικά components τότε θα βρούμε το μέγεθος του κάθε component χρησιμοποιώντας το disjoint set από το unordered map και ελέγχοντας κάθε στοιχείο του ελέγχουμε αν ανήκει σε κάποιο component από τα δύο και αυξάνουμε ένα counter για τον κάθε ένα. Έπειτα ελέγχουμε ποιο είναι το μικρότερο component ώστε να εφαρμόσουμε στον κόμβο που ανήκει σε αυτό το component την συνάρτηση evert ώστε να τον κάνουμε την ρίζα του μικρότερου component για να μπορέσουμε μετά να ενώσουμε τα δύο αυτά component κάνοντας link τους δύο κόμβους ενώνοντας έτσι τα δύο components.