ΔΙΚΤΥΑ1

ΕΡΓΑΣΙΑ ΑΝΑΠΤΥΞΗΣ ΛΟΓΙΣΜΙΚΟΥ

ΠΑΝΑΓΙΩΤΑ ΜΑΥΡΟΕΙΔΗ

Ε17098

Περιγραφή.

❑ Ανάπτυξη λογισμικού το οποίο θα υλοποιεί τους αλγορίθμους δρομολόγησης link-state (Dijkstra) και distance vector (Bellman-Ford).

❑ Το πρόγραμμα θα πρέπει να είναι εφαρμόσιμο στο δίκτυο του πάνω σχήματος – Κόμβοι: 6, Σύνδεσμοι: όπως στο σχήμα – Κόστος συνδέσμων: μεταβλητό (το σχήμα δεόχνει μία περίπτωση)

❑ Ο χρήστης θα επιλέγει τον κόμβο «αφετηρία» για τον οποίο ψάχνουμε τον πίνακα δρομολόγησης

❑ Το λογισμικό θα δίνει τον πίνακα δρομολόγησης του επιλεγμένου κόμβου αφετηρία

Είσοδος

❑ Το πρόγραμμα πρέπει να αποθηκεύει από αρχείο την τοπολογία του δικτύου και τα κόστη των συνδέσμων.

❑ Θέματα τα οποία πρέπει να ύπάρξει απόφαση: – Μορφή εισόδου. – Δομές για την εσωτερική αναπαράσταση της εισόδου.

– Ο τρόπος με τον οποίο ο χρήστης θα επιλέγει τον κόμβο

– αφετηρία των αλγορίθμων δρομολόγησης. Κυρίως Πρόγραμμα

❑ Με βάση την είσοδο, δηλαδή το δίκτυο και κάποιον επιλεγμένο κόμβο – αφετηρία, το πρόγραμμα θα πρέπει να βρίσκει τις ελάχιστες διαδρομές με βάση τους αλγορίθμους δρομολόγησης του Dijkstra και Bellman-Ford.

Έξοδος

❑ Το πρόγραμμα θα πρέπει να παρουσιάζει τον πίνακα δρομολόγησης του επιλεγμένου κόμβου αφετηρία, καθώς και τα ενδιάμεσα βήματα εκτέλεσης των δύο αλγορίθμων (πως καταλήγουν οι αλγόριθμοι στον πίνακα δρομλόγησης)

Dijkstra:

Η βασική ιδέα του αλγορίθμου Dijkstra είναι η συνεχής εξάλειψη μακρύτερων διαδρομών μεταξύ του κόμβου εκκίνησης και όλων των πιθανών προορισμών.

Για να παρακολουθούμε τη διαδικασία, πρέπει να έχουμε δύο ξεχωριστά σύνολα κόμβων, διακανονισμένα και άστατα.

Οι εγκατεστημένοι κόμβοι είναι αυτοί με γνωστή ελάχιστη απόσταση από την πηγή. Το μη ρυθμισμένο σύνολο κόμβων συγκεντρώνει κόμβους στους οποίους μπορούμε να φτάσουμε από την πηγή, αλλά δεν γνωρίζουμε την ελάχιστη απόσταση από τον αρχικό κόμβο.

Ουσιαστικα το προγραμμα ακολουθει τα εξης βηματα:

0)Ορίζουμε την απόσταση στο startNode στο μηδέν.

1)Ορίζουμε όλες τις άλλες αποστάσεις σε μια άπειρη τιμή.

2)Προσθέτουμε το startNode στο σύνολο των μη ρυθμισμένων κόμβων.

3)Ενώ το μη ρυθμισμένο σύνολο κόμβων δεν είναι κενό εμείς:

4)Επιλεγουμε έναν κόμβο αξιολόγησης από το μη ρυθμισμένο σύνολο κόμβων, ο κόμβος αξιολόγησης πρέπει να είναι αυτός με τη χαμηλότερη απόσταση από την πηγή.

5)Υπολογίζουμε νέες αποστάσεις για να κατευθύνουμε τους γείτονες διατηρώντας τη χαμηλότερη απόσταση σε κάθε αξιολόγηση.

6)Προσθέτουμε γείτονες που δεν έχουν ακόμη εγκατασταθεί στο μη ρυθμισμένο σύνολο κόμβων.

7) Ζηταμε απο τον χρήστη να μας δωσει μια τιμή κ αν αυτη ειναι απο 1 εως 6 εμφανιζει τον αλγοριθμο απο το συγκεκριμενο σημειο

Ενδεικτικο αποτελεσμα του αλγοριθμου με χρηση της java:

```
give me a number from 1 to 6

3
2 [Node [name=3]]
4 [Node [name=3]]
3 []
6 [Node [name=3]]
5 [Node [name=3]]
1 []

2 [Node [name=3]]
4 [Node [name=3]]
3 []
6 [Node [name=3]]
5 [Node [name=3]]
1 []

2 [Node [name=3], Node [name=5]]
4 [Node [name=3], Node [name=5]]
3 []
6 [Node [name=3]]
5 [Node [name=3]]
1 []

2 [Node [name=3], Node [name=5]]
4 [Node [name=3], Node [name=5]]
3 []
6 [Node [name=3]]
5 [Node [name=3]]
1 [Node [name=3], Node [name=5], Node [name=4]]

2 [Node [name=3], Node [name=5]]
4 [Node [name=3], Node [name=5]]
3 []
6 [Node [name=3]]
5 [Node [name=3]]
1 [Node [name=3], Node [name=5], Node [name=4]]

2 [Node [name=3], Node [name=5]]
4 [Node [name=3], Node [name=5]]
3 []
6 [Node [name=3]]
5 [Node [name=3]]
1 [Node [name=3], Node [name=5], Node [name=4]]
```

Bellman:

Αυτός ο αλγόριθμος υπολογίζει τις μικρότερες διαδρομές από κάτω προς τα πάνω. Υπολογίζει πρώτα τις μικρότερες αποστάσεις που έχουν το πολύ ένα άκρο στο μονοπάτι. Στη συνέχεια, υπολογίζει τις μικρότερες διαδρομές με το πολύ 2 άκρα και ούτω καθεξής. Μετά την 1η επανάληψη του εξωτερικού βρόχου, υπολογίζονται οι μικρότερες διαδρομές με τις περισσότερες άκρες. Μπορεί να υπάρχει μέγιστο | V | - 1 άκρη σε οποιαδήποτε απλή διαδρομή, γι 'αυτό τρέχει ο εξωτερικός βρόχος | v | - 1 φορές. Η ιδέα είναι, υποθέτοντας ότι δεν υπάρχει αρνητικός κύκλος βάρους, εάν έχουμε υπολογίσει τις μικρότερες διαδρομές με τις περισσότερες άκρες, τότε μια επανάληψη σε όλες τις άκρες εγγυάται την παροχή μικρότερης διαδρομής με τις άκρες το πολύ (i + 1)