Άσκηση 2 [05.05.2022]

Παράδοση: 25.05.2022, 23:59

Η παρούσα άσκηση αφορά τη σύγκριση του αλγορίθμου **Dijkstra** με μία παραλλαγή του, τον αλγόριθμο $\mathbf{A}^*$, για την επίλυση του προβλήματος της εύρεσης της συντομότερης διαδρομής μεταξύ δύο κόμβων s και t σε ένα γράφημα $G = (V, E)$, με κόστη ακμών $wt(i, j) \geq 0$, $\forall(i, j) \in E$. Σκοπός της άσκησης είναι η εξοικείωση με τη χρήση της βιβλιοθήκης **Boost**.

Για το συκεκριμένο πρόβλημα (εύρεσης της συντομότερης s-t διαδρομής) ο αλγόριθμος **Dijkstra** θα τερματίζει, μόλις βρεθεί και διαγραφεί από την ουρά προτεραιότητας ο κόμβος t, αφού δεν ενδιαφερόμαστε για τις συντομότερες διαδρομές απο τον κόμβο s προς όλους τους κόμβους του γραφήματος. Ονομάζουμε την παραλλαγή αυτή **Dijkstra-SP**.

Έστω $d(x, y)$ η απόσταση (κόστος συντομότερης διαδρομής) από έναν κόμβο x σε έναν κόμβο y. Τότε, είναι πολύ εύκολο να αποδειχθεί ότι για οποιαδήποτε τριάδα κόμβων $x, y, z \in V$ ισχύει η *τριγωνική ανισότητα*: $d(x, y) + d(y, z) \geq d(x, z)$.

Ο αλγόριθμος $\mathbf{A}^*$ βασίζεται στον υπολογισμό (σε μια προγενέστερη φάση) κάτω φραγμάτων των αποστάσεων των κόμβων προς τον t. Έστω $h_t(i)$ ένα κάτω φράγμα στην απόσταση $d(i, t)$ του κόμβου i από τον t τέτοιο, ώστε να ισχύει η σχέση $h_t(i) \leq h_t(j) + wt(i, j)$, $\forall(i, j) \in E$. Παρακάτω παρουσιάζονται δύο ενδεικτικοί τρόποι υπολογισμού του κάτω φράγματος $h_t(i)$, για οποιονδήποτε κόμβο $i \in V$.

1. Στην περίπτωση γραφήματος στο επίπεδο, αν οι κόμβοι αποτελούν σημεία του επιπέδου με συντεταγμένες (x_i, y_i) και το κόστος κάθε ακμής ισούται με την Ευκλείδια απόσταση μεταξύ των άκρων της, τότε προφανώς η ποσότητα $h_t(i) = \sqrt{(x_i - x_t)^2 + (y_i - y_t)^2}$ αποτελεί κάτω φράγμα στην απόσταση $d(i, t)$.

2. Στην περίπτωση τυχαίου γραφήματος, επιλέγεται αυθαίρετα ένας κόμβος ορόσημο L_1, και υπολογίζονται οι συντομότερες διαδρομές από τον L_1 προς όλους τους υπόλοιπους κόμβους του γραφήματος (καλώντας τον αλγόριθμο **Dijkstra**) και αντίστροφα, δηλαδή από όλους τους υπόλοιπους κόμβους του γραφήματος προς τον L_1 (καλώντας τον αλγόριθμο **Dijkstra** με τη διαφορά ότι, αντί να εξετάζονται προς χαλάρωση οι εξερχόμενες ακμές του κόμβου v που διαγράφηκε από την ουρά προτεραιότητας, εξετάζονται οι εισερχόμενες ακμές του v).

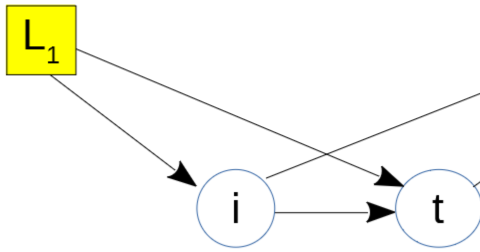
   Στη συνέχεια επιλέγεται ένας ακόμα κόμβος ορόσημο L_2, ο οποίος βρίσκεται όσο το δυνατόν μακρύτερα από τον L_1, και με παρόμοιο τρόπο υπολογίζονται οι συντομότερες διαδρομές από τον L_2 προς όλες τις κορυφές του γραφήματος και αντίστροφα.

   Εφαρμόζοντας την τριγωνική ανισότητα στα ορόσημα L_1 και L_2 (δείτε και Σχήμα 1), προκύπτει ότι:

   $$d(L_1, i) + d(i, t) \geq d(L_1, t)$$
   $$d(i, t) + d(t, L_2) \geq d(i, L_2)$$

   Λύνοντας και τις δυο ανισότητες ως προς $d(i, t)$ προκύπτει:

   $$d(i, t) \geq d(L_1, t) - d(L_1, i)$$



Σχήμα 1

$$d(i,t) \geq d(i,L_2) - d(t,L_2)$$

Το (καλύτερο) κάτω φράγμα $h_t(i)$ της κορυφής i από την t είναι το μέγιστο των δυο αυτών όρων.

$$h_t(i) = \max\{d(L_1,t) - d(L_1,i), d(i,L_2) - d(t,L_2)\}$$

Σημειώνεται ότι με τον ίδιο τρόπο μπορούν να επιλεγούν και άλλα τέτοια ορόσημα και στην περίπτωση αυτή το προηγούμενο μέγιστο αφορά όλα αυτά τα ορόσημα.

Ο αλγόριθμος $\mathbf{A}^*$ αποτελεί μια παραλλαγή του αλγορίθμου **Dijkstra**, η οποία χρησιμοποιεί τη συνάρτηση h_t για να μεταβάλει αρχικά το κόστος κάθε ακμής $(i,j) \in E$ σε $wt'(i,j) = wt(i,j) + h_t(j) - h_t(i)$, και κατόπιν ακολουθεί τη μέθοδο **Dijkstra-SP**. Έστω $d'(s,t)$ η απόσταση του κόμβου t από τον s στο γράφημα με συνάρτηση κόστους ακμών $wt'(\cdot)$. Είναι εύκολο να δείτε ότι $d'(s,t) = d(s,t) + h_t(t) - h_t(s)$. Αφού $h_t(t) = 0$, η ζητούμενη απόσταση $d(s,t)$ προκύπτει από την τιμή $d'(s,t)$ που έχει υπολογίσει ο $\mathbf{A}^*$ προσαυξημένη κατά $h_t(s)$.

Στην άσκηση αυτή ζητείται να υλοποιήσετε τους αλγορίθμους **Dijkstra-SP** και $\mathbf{A}^*$, όπως περιγράφηκαν παραπάνω, με χρήση της βιβλιοθήκης **Boost**, και να τους συγκρίνετε πειραματικά τόσο ως προς το χρόνο εκτέλεσης, όσο και ως προς τον αριθμό των κόμβων που εξετάζουν, μέχρι να οριστικοποιηθεί η απόσταση $d(s,t)$ προς τον t. Ο υπολογισμός της συνάρτησης h_t θα πρέπει να γίνει σε ένα στάδιο προεπεξεργασίας, δηλαδή πριν εκτελεστεί ο αλγόριθμος $\mathbf{A}^*$ (ο χρόνος προεπεξεργασίας δεν θα προσμετράται στον χρόνο εκτέλεσης του $\mathbf{A}^*$).

Η πειραματική αξιολόγηση πρέπει να διεξαχθεί:

- Σε γραφήματα τύπου πλέγματος (grid) μεγέθους $r \times c$ (γραμμές $\times$ στήλες), όπου $(r,c) \in \{(30,1000),(60,1000),(80,1000)\}$. Θεωρούμε ότι ο πάνω αριστερά κόμβος έχει συντεταγμένες $(0,0)$ και ο κάτω δεξια $(r-1,c-1)$. Η αρχική κορυφή s θα επιλέγεται τυχαία από τη στήλη 0 και η τελική κορυφή t θα επιλέγεται τυχαία από τη στήλη $c-1$. Για την κατευθυνόμενη εκδοχή (την οποία καλείστε να υλοποιήσετε) υποθέτουμε ότι για κάθε ακμή υπάρχει και η αντίθετή της με το ίδιο κόστος, το οποίο είναι ένας τυχαίος ακέραιος αριθμός στο διάστημα $[1,2]$.

- Σε τυχαία γραφήματα n κορυφών και m ακμών, όπου $(n, m) \in \{(10.000, 20.000), (20.000, 40.000), (60.000, 120.000)\}$.

Και για τις δύο κατηγορίες γραφημάτων, θα εξετάσετε δυο περιπτώσεις κοστών ακμών : α) να έχουν τυχαίες ακέραιες τιμές στο διάστημα $[1, 100]$ και β) να έχουν τυχαίες ακέραιες τιμές στο διάστημα $[1, 10.000]$.

Bonus: Θα δοθεί επιπλέον βαθμολογία σε όποιον/α:

- Δημιουργήσει μια συνάρτηση η οποία μετατρέπει το γράφημα με τα νέα κόστη ακμών από Boost σε Leda.
- Αξιολογήσει πειραματικά τον αλγόριθμο της Leda NT DIJKSTRA_T* (μόνο ως προς το χρόνο) στο παραπάνω γράφημα.

*την παραλλαγή που δέχεται ως είσοδο κόμβο αφετηρίας και κόμβο προορισμού.

Υποβολή Προγραμματιστικής Άσκησης:

Η παράδοση της εργασίας θα πραγματοποιηθεί ηλεκτρονικά, μέσω της σελίδας του μαθήματος στο eclass, υποβάλοντας ένα συμπιεσμένο αρχείο που περιέχει όλα τα παραδοτέα της εργασίας:

1. **report.pdf**
   Ένα pdf αρχείο που περιέχει την αναφορά της εργασίας σας, στην οποία περιγράφονται οι βασικές αποφάσεις της υλοποίησής σας, τα δεδομένα δοκιμής που χρησιμοποιήσατε και η πειραματική αξιολόγηση.
2. **Makefile**
   Το αρχείο Makefile που χρησιμοποιήσατε.
3. **src/**
   Ένα φάκελο που θα περιέχει όλα τα αρχεία του πηγαίου κώδικά σας. Ο πηγαίος κώδικας που δίνετε για τις υλοποιήσεις και πειραματικές αξιολογήσεις πρέπει να είναι σωστά δομημένος, στοιχισμένος και σχολιασμένος. Ο κώδικάς σας πρέπει να εκτελείται στο σύστημα diogenis.
4. **README**
   Ένα αρχείο που θα περιέχει τα στοιχεία σας (Όνομα, Επώνυμο, ΑΜ, email)

Παρατήρηση 1: Μπορείτε να χρησιμοποιήσετε όποιους τύπους της BOOST κρίνετε απαραίτητο.

Παρατήρηση 2: Για περαιτέρω διευκρινήσεις σχετικά με την άσκηση, μπορείτε να επικοινωνήσετε με τον Νίκο Ζαχαράτο (zacharato@ceid.upatras.gr) ή τη Βούλα Μαχαίρα (machaira@ceid.upatras.gr).